

# Desarrollo infraestructura tipo Cloud mediante tecnología Ignition



Máster Universitario en Automática  
y Robótica

## Trabajo Fin de Máster

Autor:

Lachguer Abdallah

Tutor/es:

Carlos Alberto Jara Brava

Víctor García García



Universitat d'Alacant  
Universidad de Alicante

Julio 2020



# Desarrollo infraestructura tipo Cloud mediante tecnología Ignition

---

**Autor**

Lachguer Abdallah

**Tutor/es**

Carlos Alberto Jara Brava

Víctor García García

**Departamento**

Física, Ingeniería de Sistemas y Teoría de la Señal



Máster Universitario en Automática y Robótica

ALICANTE, julio 2020



Escuela  
Politécnica  
Superior



Universitat d'Alacant  
Universidad de Alicante



# Resumen

El presente trabajo fin de master “Desarrollo infraestructura tipo *cloud* mediante tecnología *Ignition*”, comprende el estudio de la posibilidad de implementar la tecnología *Ignition* en la empresa Suez, debido a que la plataforma actual llevada a cabo en la empresa para sistemas de SCADA (*Supervisory Control And Data Adquisition*) está basada únicamente en la tecnología *Wonderware*, cosa que a nivel operativo Suez estimó que tener una única tecnología SCADA suponía un riesgo operativo. La plataforma *Ignition* es la principal línea de productos de la compañía *Inductive Automation*.

El objetivo principal es mitigar el riesgo operativo de dependencia de la empresa SUEZ ante un único fabricante de plataforma corporativa de gestión de centros de control operativos, para poder así ofrecer a otros clientes, o migrar las aplicaciones SCADA ya existentes a una plataforma con una tecnología distinta y económica.

Para llevar a cabo lo descrito anteriormente, se analizará la viabilidad de creación de una plataforma con la tecnología *Ignition*. y para ello se descompondrán los diferentes módulos, plantillas, funcionalidad, etc. de la aplicación *Wonderware*, luego se realizará un análisis y descripción técnica de alto nivel de como cada una de las partes descompuestas de *Wonderware* y comprobar si puede ser cubierta con la plataforma *Ignition*, y en caso contrario se describirán posibles soluciones, finalmente se hará un presupuesto de creación de dicha plataforma.

Palabras clave:

- SCADA
- IGNITION
- MONITORIZACIÓN
- SUPERVISIÓN
- HMI



# Abstract

This master thesis "Development of cloud-type infrastructure using Ignition technology" includes the study of the possibility of implementing Ignition technology in the company SUEZ, due to the fact that the current platform carried out in the company for SCADA systems (Supervisory Control And Data Acquisition) is based solely on Wonderware technology, which at an operational level Suez estimated that having a single SCADA technology posed an operational risk. The Ignition platform is the main product line of the Inductive Automation company.

The main objective is to mitigate the operational risk of dependency of the SUEZ company before a single manufacturer of a corporate platform for managing operational control centers, in order to be able to offer other clients, or migrate existing SCADA applications to a platform with a technology different, and economic.

To carry out the previously described, the feasibility of creating a platform with Ignition technology will be analyzed. and for this the different modules, templates, functionality, etc. will be broken down. of the Wonderware application, then a high-level analysis and technical description of how each of the decomposed parts of Wonderware will be carried out and check if it can be covered with the Ignition platform, and if not, possible solutions will be described, finally it will be done a budget for the creation of said platform.

Keywords:

- SCADA
- IGNITION
- MONITORING
- SUPERVISION
- HMI





# Agradecimientos

A Víctor García García y a Jorge Boluda Segura, darles las gracias por su tiempo, dedicación y conocimientos que me han ofrecido desde el primer día. Me han enseñado mucho y sin ellos no hubiera sido posible realizar este proyecto.

A mi profesor Carlos Jara por haberme ayudado a realizar este proyecto y por las correcciones y consejos de mejora que me ha dado para redactar la memoria.

A la Universidad de Alicante, a la Escuela Politécnica Superior y a todos mis profesores del máster universitario en Automática y Robótica.

A la empresa Aquatec por permitirme hacer el proyecto en sus instalaciones.

A mi amiga y compañera del máster Mercedes Navarrate, y a mis amigos Mohamed Hnina, El Habib Majidi por estar siempre ahí y por su apoyo constante.



*Dedicado a mis padres y a mi familia por su apoyo y confianza durante todos  
mis años de estudio.*



*Mejor no hacer nada que hacer cualquier cosa.*

Francis Picabia



# Índice de contenidos

<b>1</b>	<b>CAPÍTULO 1: INTRODUCCIÓN .....</b>	<b>1</b>
1.1	Antecedentes .....	1
1.2	Objetivo general.....	2
1.3	Objetivos específicos .....	2
1.4	Metodología .....	2
1.5	Estructura de la memoria .....	3
<b>2</b>	<b>CAPÍTULO 2: ESTADO DE ARTE.....</b>	<b>5</b>
2.1	El Sistema SCADA .....	5
2.1.1	Definición .....	5
2.1.2	Beneficios.....	6
2.1.3	Arquitectura.....	6
2.1.4	Componentes de un sistema SCADA .....	8
2.1.5	Comunicación entre aplicaciones.....	8
2.2	El Software Ignition .....	8
2.2.1	Introducción.....	8
2.2.2	Gateway Webpage.....	9
2.2.3	Módulos .....	13
2.2.4	Arquitectura.....	15
2.2.5	Comparativa entre <i>Ignition</i> y otras plataformas .....	16
2.3	Criterios y requisitos de diseño de la aplicación SCADA .....	17
2.3.1	Criterios.....	17
2.3.2	HMI alto rendimiento .....	18
2.3.3	La norma ISA101 .....	19
2.3.4	Requisitos.....	19
2.3.4.1	Análisis de la plataforma Suez .....	20
<b>3</b>	<b>CAPÍTULO 3: IMPLEMENTACIÓN.....</b>	<b>27</b>
3.1	Instalación de software <i>Ignition</i> .....	27
3.2	Conexión con la base de datos .....	28
3.3	Creación de un proyecto.....	30

3.4	Creación de ventanas .....	31
3.5	Creación de UDTs .....	32
3.6	Creación de <i>Templates</i> .....	34
3.7	Scripting en Ignition .....	36
3.8	Aplicación SCADA.....	36
3.8.1	UDTs Creados.....	36
3.8.2	InstanciasCreadas .....	37
3.8.3	Descripción da tags .....	38
3.8.4	Tabla de señales.....	38
3.8.5	<i>Templates</i> creados.....	39
3.8.6	Interfaz de la aplicación .....	39
3.8.7	Gestión de históricos .....	49
3.8.8	Gestión de Alarmas.....	51
3.8.9	Notificación de alarmas .....	57
4	<b>Capítulo 4: Presupuesto, Conclusiones y Trabajos futuros .....</b>	<b>63</b>
4.1	<b>Presupuestos .....</b>	<b>63</b>
4.1.1	Presupuesto Mínimo Producto Viable.....	63
4.1.2	Presupuesto desarrollo de una plataforma global en <i>Ignition</i> .....	64
4.2	<b>Conclusiones y Trabajos futuros.....</b>	<b>65</b>
5	<b><i>Bibliografía</i>.....</b>	<b>67</b>
6	<b><i>Anexos</i>.....</b>	<b>69</b>
	Anexo A: Código del generador de instancias .....	69
	Anexo B: Código de descripciones de <i>tags</i> .....	71
	Anexo C: Lista de señales del proyecto .....	73
	Anexo D: Lista de <i>templates</i> del proyecto.....	77
	Anexo E: Código trazador de variables.....	83
	Anexo F: Código Integración de <i>dataloggers</i> .....	85



# Índice de figuras

<b>Ilustración 1:</b> Cronograma de las tareas del proyecto	2
<b>Ilustración 2:</b> Pirámide de la Automatización Industrial	6
<b>Ilustración 3:</b> Estructura Básica de un Sistema de Supervisión y Mando	7
<b>Ilustración 4:</b> Arquitectura básica de Hardware de un SCADA	7
<b>Ilustración 5:</b> Bloque Home del Gateway	10
<b>Ilustración 6:</b> Bloque Status del Gateway	11
<b>Ilustración 7:</b> Bloque Configuration del Gateway	12
<b>Ilustración 8:</b> Arquitectura del Gateway	13
<b>Ilustración 9:</b> Un mapa creado con el módulo de Kyemra Map Panel	15
<b>Ilustración 10:</b> HMI tradicional (a) y HMI alto rendimiento (b)	18
<b>Ilustración 11:</b> Ejemplo de un sinóptico	20
<b>Ilustración 12:</b> Ejemplo de una ventana de estación	21
<b>Ilustración 13:</b> Faceplate de una bomba	21
<b>Ilustración 14:</b> Barra superior de la aplicación Wonderware	22
<b>Ilustración 15:</b> Barra lateral de la aplicación Wonderware	22
<b>Ilustración 16:</b> Árbol de Sinópticos General	22
<b>Ilustración 17:</b> Buscador de estaciones	23
<b>Ilustración 18:</b> Ventana de históricos genéricos de la aplicación Wonderware	24
<b>Ilustración 19:</b> Ventana de gestión de alarmas de la aplicación Wonderware	24
<b>Ilustración 20:</b> Bomba en Alarma	25
<b>Ilustración 21:</b> Página de descarga de Ignition	27
<b>Ilustración 22:</b> Página de Registro en la página de Inductive Automation	27
<b>Ilustración 23:</b> Conexión a una base de datos	29
<b>Ilustración 24:</b> Crear una nueva conexión a la base de datos	29
<b>Ilustración 25:</b> Seleccionar la base de datos	29
<b>Ilustración 26:</b> Descarga de "El diseñador"	30
<b>Ilustración 27:</b> Crear un nuevo proyecto	31
<b>Ilustración 28:</b> Datos de un proyecto nuevo	31
<b>Ilustración 29:</b> Crear una nueva ventana	32
<b>Ilustración 30:</b> Crear un nuevo UDT	32
<b>Ilustración 31:</b> Agregar un Tag al UDT	33
<b>Ilustración 32:</b> Exportar/ importar UDT	33
<b>Ilustración 33:</b> Crear un nuevo Template	34
<b>Ilustración 34:</b> Property Editor de un Template	34
<b>Ilustración 35:</b> Acceso a Custom Properties de un template	35
<b>Ilustración 36:</b> Custom Properties de un template	35

<b>Ilustración 37:</b> Estructura de carpetas del proyecto	37
<b>Ilustración 38:</b> Generador de Instancias	37
<b>Ilustración 39:</b> Interfaz de la aplicación desarrollada	39
<b>Ilustración 40:</b> Ventana superior de la aplicación desarrollada	39
<b>Ilustración 41:</b> Ventana de navegación	40
<b>Ilustración 42:</b> Información del usuario conectado	41
<b>Ilustración 43:</b> Controles de la aplicación	41
<b>Ilustración 44:</b> Ventana de cambio de usuario	41
<b>Ilustración 45:</b> Ventana de bloqueo de la aplicación	42
<b>Ilustración 46:</b> Mensaje de confirmación de cierre de sesión	42
<b>Ilustración 47:</b> Información sobre los botones de la barra superior	42
<b>Ilustración 48:</b> Ventana inferior de la aplicación desarrollada.	43
<b>Ilustración 49:</b> Panel de comentarios	43
<b>Ilustración 50:</b> Ventana principal de la aplicación desarrollada	44
<b>Ilustración 51:</b> Fallo de una bomba	44
<b>Ilustración 52:</b> Estación la Drova	45
<b>Ilustración 53:</b> Ventana de estación	45
<b>Ilustración 54:</b> Trazador de variables	46
<b>Ilustración 55:</b> Acceso al faceplate de bomba	46
<b>Ilustración 56:</b> Faceplate de una Bomba	47
<b>Ilustración 57:</b> Pestaña alarmas del faceplate de Bomba	47
<b>Ilustración 58:</b> Indicación de alarma en un faceplate de Bomba	48
<b>Ilustración 59:</b> Pestaña Control del faceplate Bomba	48
<b>Ilustración 60:</b> Mensaje de confirmación	48
<b>Ilustración 61:</b> Acceso a la ventana de históricos	49
<b>Ilustración 62:</b> Ventana de históricos de la aplicación desarrollada	49
<b>Ilustración 63:</b> Acceso a la ventana de alarmas	51
<b>Ilustración 64:</b> Ventana de alarmas de la aplicación desarrollada	51
<b>Ilustración 65:</b> Tabla de alarmas en estado real	52
<b>Ilustración 66:</b> Personalizar la tabla de alarmas	54
<b>Ilustración 67:</b> Opciones de la tabla de alarmas	54
<b>Ilustración 68:</b> Detalles de alarma seleccionada	55
<b>Ilustración 69:</b> Histórico de la alarma seleccionada	55
<b>Ilustración 70:</b> Ventana de "Shelved Alarms"	55
<b>Ilustración 71:</b> Tabla de históricos de alarmas	56
<b>Ilustración 72:</b> Ventana de filtros de la tabla de alarma	56
<b>Ilustración 73:</b> Notificación por correo electrónico	58
<b>Ilustración 74:</b> Configuración del servidor SMTP	58

<b>Ilustración 75:</b> Configuración del grupo de usuarios	58
<b>Ilustración 76:</b> Añadir usuario al grupo de usuarios	59
<b>Ilustración 77:</b> Información de correo electrónico de un usuario	59
<b>Ilustración 78:</b> Crear un nuevo New Pipeline	59
<b>Ilustración 79:</b> Bloques de funciones en la lógica de notificación	60
<b>Ilustración 80:</b> Lógica de notificación	60
<b>Ilustración 81:</b> Configuración de tags con el Pipeline	60
<b>Ilustración 82:</b> Recepción de correo de notificación de alarmas	61



# Índice de tablas

<b>Tabla 1:</b> Arquitecturas de Ignition.....	16
<b>Tabla 2:</b> Comparativa Ignition y otras plataformas.....	17
<b>Tabla 3:</b> Datos de conexión a la base de datos .....	30
<b>Tabla 4:</b> UDTs Creados .....	36
<b>Tabla 5:</b> Tablas de las instancias del proyecto.....	38
<b>Tabla 6:</b> Elementos de navegación de la ventana superior.....	40
<b>Tabla 7:</b> Elementos adicionales de la ventana superior.....	40
<b>Tabla 8:</b> Presupuesto elaboración TFM.....	63
<b>Tabla 9:</b> Presupuesto desarrollo de una plataforma global con Ignition.....	64



# 1 CAPÍTULO 1: INTRODUCCIÓN

## 1.1 Antecedentes

Suez es una empresa que opera principalmente en los sectores de tratamiento de aguas, gestión de residuos y mantenimiento.

Las actividades de Suez incluyen:

- La producción y distribución de agua potable.
- El tratamiento de agua residual.
- Diseño y fabricación de plantas de depuración de aguas.
- Gestión de agua residual doméstica e industrial.

Suez también opera en el dominio de las plantas desaladoras de agua. La compañía organiza sus actividades en torno a un número de temas, incluidas la protección de costes, el suministro de aguas a zonas sin acceso, la distribución de agua de fuentes alternativas, la gestión del impacto industrial y la descontaminación de suelos.

Para la gestión de residuos, Suez realiza las siguientes actividades:

- Ordena y procesa desechos domésticos, industriales, médicos y agrícolas.
- Recicla e incinera residuos y los dispone en almacenes de gestión.
- Realiza la descontaminación de tierras contaminadas.

Suez dispone de un módulo SCADA para controlar los procesos en tiempo real. Se trata de un sistema estándar basado en la herramienta *System Platform* de *Wonderware*.

*Ignition* de *Automation Inductive* es la nueva plataforma SCADA que Suez desea implementar. Se trata de la principal línea de productos de la compañía *Automation Inductive*, proveedor de software de automatización industrial basado en la web con sede en California, EEUU.

*Ignition* es una plataforma de software integrada para sistemas SCADA, utilizada para lograr una implementación basada en web multiplataforma a través de la tecnología *Java* tanto para clientes en tiempo de ejecución como para el entorno de desarrolladores.

## 1.2 Objetivo general

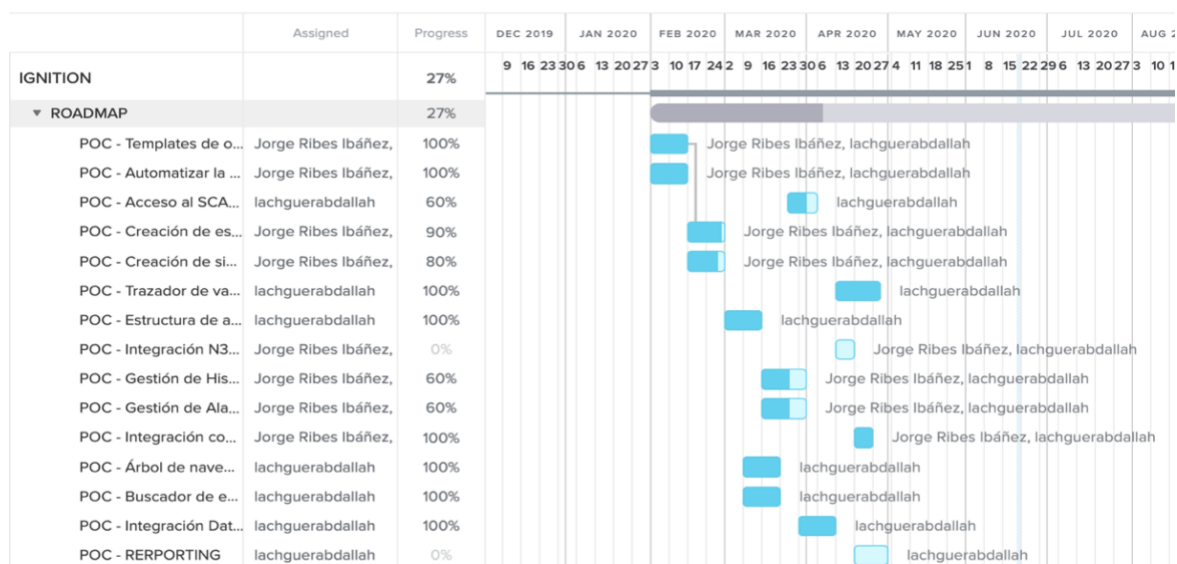
El objetivo principal de este trabajo fin de máster es desarrollar una aplicación SCADA con la tecnología *Ignition*. La aplicación SCADA que se va a crear permitirá a la empresa SUEZ comprobar si dicha tecnología podrá ser usada en la gestión de centros de control operativos.

## 1.3 Objetivos específicos

- Estudiar las características de la tecnología *Ignition*.
- Estudiar las ventajas de la tecnología *Ignition* frente a la tecnología *Wonderware* usada en la empresa Suez.
- Desarrollar una aplicación SCADA basada en la tecnología *Ignition*.
- Probar la aplicación desarrollada y comprobar si cumple con los requisitos principales.

## 1.4 Metodología

- Se desarrolla principalmente, en las oficinas de SUEZ, con el objetivo de entender la problemática y de la resolución de dudas.
- La dirección del proyecto, sobre todo inicialmente, los pasos a seguir y de las tareas a realizar durante el proyecto fueron guiadas por SUEZ, para ello, además de reuniones periódicas, se estableció el cronograma mostrado en la ilustración 1.



*Ilustración 1: Cronograma de las tareas del proyecto*



## 1.5 Estructura de la memoria

La memoria va a constar de tres capítulos más.

En el Capítulo 2, se explicarán los materiales y los métodos utilizados para elaborar este trabajo. Se hablará de los diferentes conceptos de teoría del sistema SCADA, y se analizarán las características y las opciones del software *Ignition* utilizado en el presente trabajo para el desarrollo de la aplicación SCADA. También se expondrán los requerimientos y criterios que se van a tener en cuenta para el diseño de la aplicación. Además, se hará una descripción de los módulos analizados en la aplicación *Wonderware*.

Posteriormente, en el Capítulo 3 se expondrán los pasos seguidos para diseñar la aplicación SCADA, dichos pasos consisten en la instalación del *software* necesario, y el desarrollo de las actividades llevadas a cabo en el proyecto. Además, se explicarán las diferentes características de la aplicación final desarrollada.

Por último, en el Capítulo 4 se hará un presupuesto que contendrá una estimación económica para desarrollar una plataforma global SCADA en *Ignition*, y se hablará de las conclusiones que se han podido extraer del trabajo a lo largo de la realización de éste.



# 2 CAPÍTULO 2: ESTADO DE ARTE

## 2.1 El Sistema SCADA

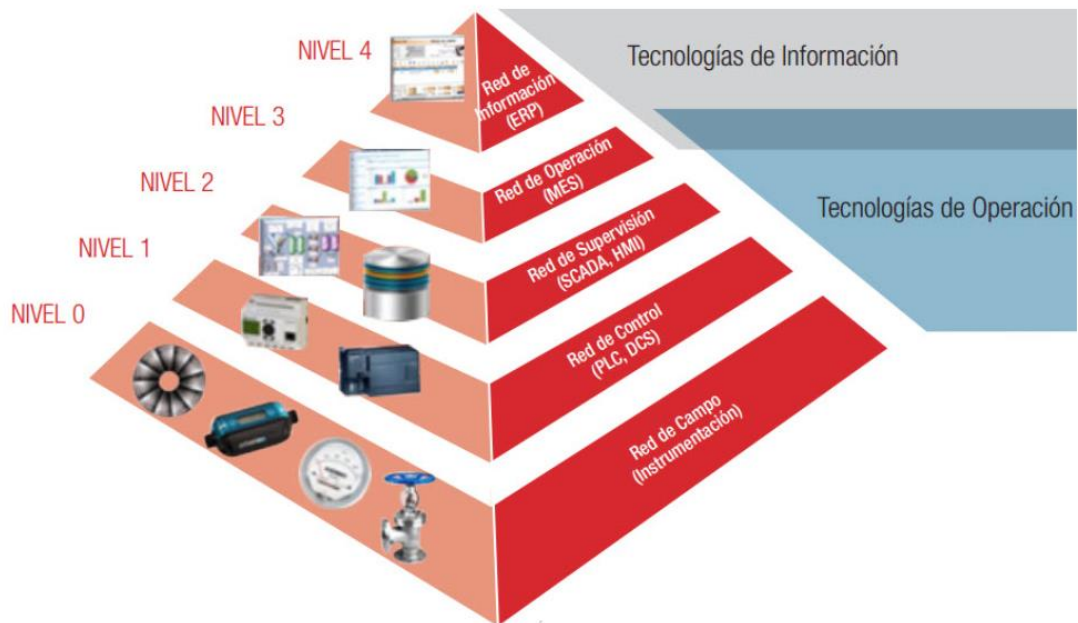
### 2.1.1 Definición

Las siglas SCADA corresponden a *Supervisory Control And Data Acquisition* o Supervisión Control y Adquisición de Datos. Un sistema SCADA permite a los operadores de un sistema industrial llevar un control completo de todos los componentes de dicho sistema.

Entre las utilidades del SCADA se encuentran (Penin, 2013):

- **Monitorización:** Representación de datos en tiempo real a los operadores.
- **Supervisión:** Supervisión, mando y adquisición de datos de un proceso y herramientas de gestión para la toma de decisiones.
- **Visualización de alarmas y eventos:** detección de situaciones o anomalías en el sistema para indicarlas y notificarlas a los operarios.
- **Mando:** o el control del sistema. Permitir a los operarios controlar los diferentes componentes de un sistema desde un ordenador. Por ejemplo: arrancar un motor, o cambiar las consignas de un subproceso.
- **Grabación de acciones o recetas:** configurar fácilmente las líneas de producción.
- **Seguridad de los datos y en los accesos:** Permite garantizar la seguridad en el proceso a controlar, registrando todos los accesos y acciones llevadas por los operadores, así como restringir el acceso a los operadores que no tienen los suficientes privilegios.
- **Programación numérica:** permite realizar cálculos aritméticos sobre la CPU del ordenador.

Los sistemas SCADA se encuentran en el nivel de Supervisión de la pirámide de automatización industrial. Tal y como se puede observar en la ilustración 2.



*Ilustración 2: Pirámide de la Automatización Industrial*  
Fuente: (Borghello, 2017)

### 2.1.2 Beneficios

Entre los beneficios de un sistema SCADA, se encuentran las siguientes (Penin, 2013):

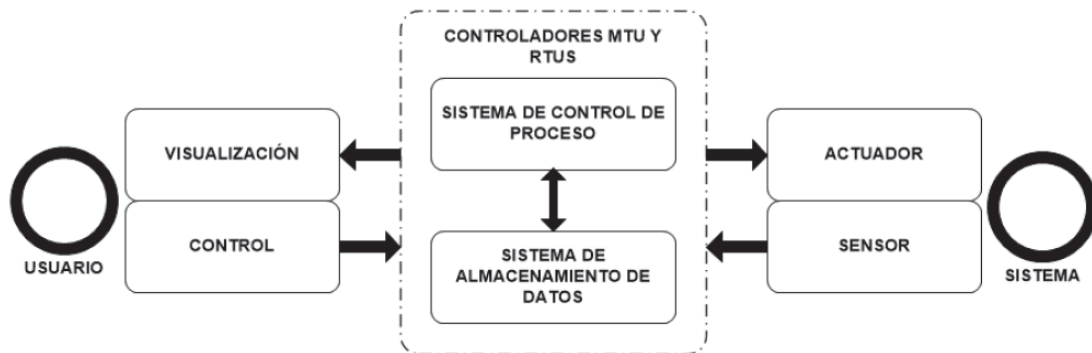
- Controlar de manera remota los diferentes elementos del proceso.
- Integración de todos los componentes del proceso.
- Alertar sobre los fallos que se producen en el proceso.
- Mostrar información real y confiable de estados de dispositivos, mediciones, etc.
- Mostrar tendencias de datos de la planta para que puedan ser procesados.
- Permitir una gestión segura y eficiente de los datos, mediante protocolos de seguridad, limitando el acceso a personas no autorizadas.

### 2.1.3 Arquitectura

Originalmente, en el campo de la automatización industrial se concentraba en una PC. Actualmente el control está formado de los siguientes componentes (Penin, 2013):

- Software de control supervisión y adquisición de datos: SCADA
- Hardware de adquisición de datos: Sensores y actuadores.
- Sistemas de interconexión: Comunicación.

La estructura básica de un sistema SCADA se representa en la ilustración 3.

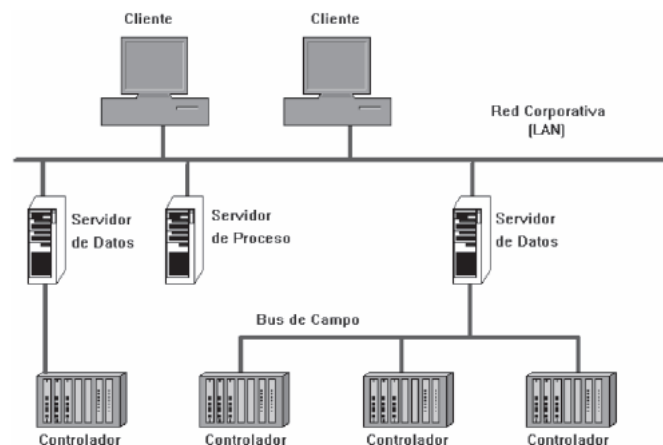


**Ilustración 3:** Estructura Básica de un Sistema de Supervisión y Mando  
Fuente: (Penin, 2013)

La arquitectura básica de hardware en un sistema SCADA se divide en los siguientes bloques (Penin, 2013):

- Recopiladores de datos: captan los datos de los elementos de un sistema, por ejemplo, PLCs, registradores, etc.
- Utilizadores de Datos: los que utilizan la información recogida por los recopiladores de datos.

La arquitectura básica de un sistema SCADA se representa en la ilustración 4.



**Ilustración 4:** Arquitectura básica de Hardware de un SCADA  
Fuente: (Penin, 2013)

### 2.1.4 Componentes de un sistema SCADA

Los componentes de hardware de un sistema SCADA son los siguientes (Penin, 2013)

- **Ordenador Central o MTU (*Master Terminal Unit*):** Se trata del ordenador principal del sistema. Tiene como objetivo supervisar y recoger información del resto de subestaciones. Interroga de forma periódica y ejecuta software especializado que cumple funciones del sistema.
- **Ordenador Remotos RTU (*Remote Terminal Unit*):** se trata de dispositivos de nivel intermedio, entre los ordenadores central y los instrumentos de campo. Los ordenadores remotos se sitúan en los nodos estratégicos de los sistemas gestionando y controlando las subestaciones.
- **Red comunicación:** una red de comunicación permite el intercambio entre la estación maestra y las unidades terminales remotas.
- **Instrumentos de campo:** consisten en los elementos que realizan el control del sistema (PLC, controladores y actuadores, así como los que captan información (sensores)

### 2.1.5 Comunicación entre aplicaciones

Los métodos de intercambio de información entre aplicaciones informáticas más conocidos son OPC, ODBC, SQL, ASCII, API

En el presente proyecto, los métodos de intercambio entre aplicaciones utilizados son:

- **OPC:** Corresponde a las siglas de *Open Protocol Communication*. Es un estándar de comunicación abierto e independiente. OPC está muy presente en la industria debido a su tecnología escalable e independiente de los fabricantes, así como el mantenimiento simple de su arquitectura en comparación con otros protocolos.
- **SQL:** Corresponde a las siglas *Structured Query Language*. Se trata de lenguaje de programación para manipulación de datos almacenados en las bases de datos.

## 2.2 El Software Ignition

### 2.2.1 Introducción

*Ignition* es un software para diseñar sistemas HMI SCADA. Es una tecnología basada en la web, en un servidor central al cual se accede mediante cualquier navegador. *Ignition* necesita la plataforma *Java* para ejecutarse, por lo tanto, puede ser utilizado en cualquier sistema operativo.

*Ignition* tiene las siguientes características (Inductive Automation, 2020):

- **Arquitectura Modular y escalable:** solo se instalan los módulos necesarios para el desarrollo de la aplicación objetivo.
- **Arquitectura Cliente Servidor:** *Ignition* está basado en un servidor, que es el que controla todas las herramientas.
- **Licencia ilimitada:** las licencias de *Ignition* tiene un coste económico, depende de los módulos que se emplean. Una ventaja que posee *Ignition* frente a otros programas es que una dispone de licencias ilimitadas y permite un acceso ilimitado de clientes, diseñadores, así como *Tags* ilimitados.
- **Servidor basado en la web:** se puede acceder al servidor de *Ignition* desde cualquier navegador.

### 2.2.2 Gateway Webpage

*Gateway Webpage* concentra todas las características, configuraciones y funcionalidades de la plataforma. Desde el *Gateway* se pueden realizar las siguientes funciones (Inductive Automation, 2020):

- Las conexiones a los dispositivos, bases de datos.
- La configuración del software.
- Activación de licencia.
- Realizar copias de seguridad.
- Instalación de módulos.
- Administración de alarmas.
- Gestión de usuarios.

Se accede a la página del *Gateway* a través de un navegador web. El navegador web que se ejecuta en cualquier máquina debe tener acceso de red al host que ejecuta el *Gateway*.

De manera predeterminada *Ignition* se instala utilizando el puerto 8088, siendo la dirección que hay que ingresar para acceder a la pagina del *Gateway* la siguiente:

**Http://ip\_host:8088**

Cualquier otro ordenador en la misma red puede acceder al *Gateway* utilizando la dirección IP o el nombre de host del ordenador donde esta instalado *Ignition*.

Se accede a la puerta de enlace a través de un navegador web. El navegador web, que se ejecuta en cualquier máquina, debe tener acceso de red al *host* que ejecuta el *Gateway*.

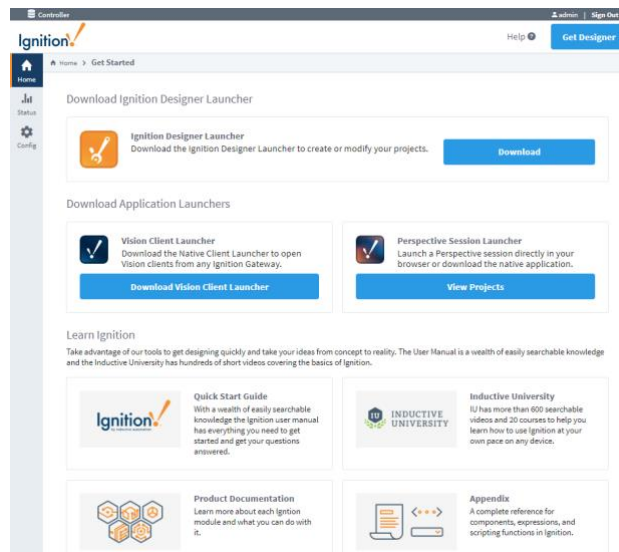
la página del Gateway se divide en tres bloques:

- *Home*
- *Status*
- *Configuration*

En los siguientes puntos se hará una descripción de cada bloque.

### **Bloque 1: *Home***

En la ilustración 5 se muestra la pagina *Home*, es la primera página que aparece cuando se inicia el *Gateway*.



***Ilustración 5: Bloque Home del Gateway***

En la página *Home* se pueden realizar muchas funciones, de las cuales se mencionan las siguientes:

- Descargar el *Designer Launcher*, es el programa que se usa para diseñar y programar las aplicaciones,
- Descargar *Vision Client Launcher*: es el programa que se usa para instalar en las maquinas de clientes.
- Descargar *Perspective sesión launcher*: es el programa que se usa para ejecutar sesiones en navegadores web.

Además de lo anterior, en la página *Home* se puede acceder a manuales de *Ignition*, otra documentación y enlaces útiles para iniciar en la plataforma.

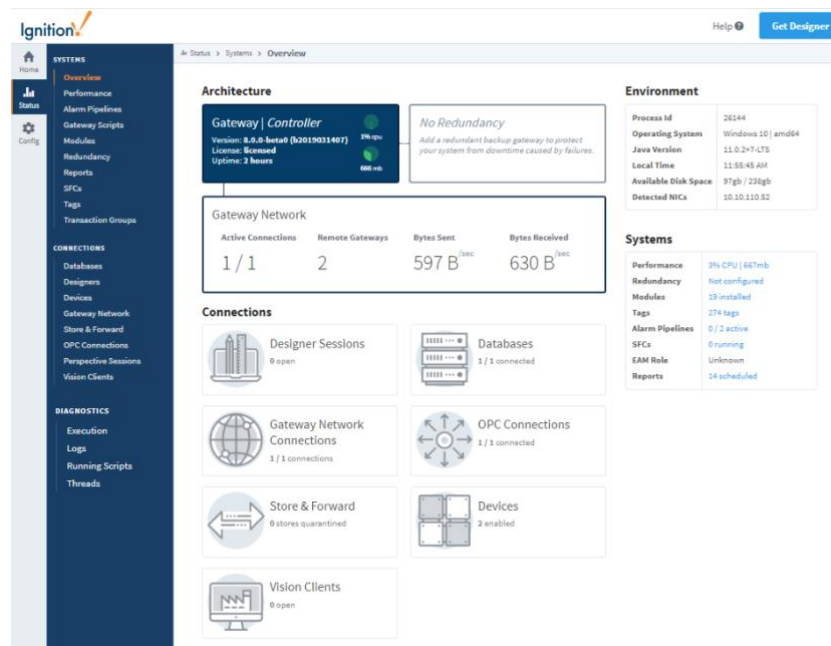


## Bloque 2: Status

La pestaña *Status* proporciona información detallada sobre el estado de las diferentes partes del sistema. Entre las informaciones que se pueden consultar, se encuentran las siguientes:

- Número y estado de clientes conectados
- Número y estado de dispositivos conectados
- Número y estado de las bases de datos
- *Logs* del sistema.

Los elementos que se muestran en la página *Status* cambian según los módulos instalados. La página *Status* se muestra en la ilustración 6.

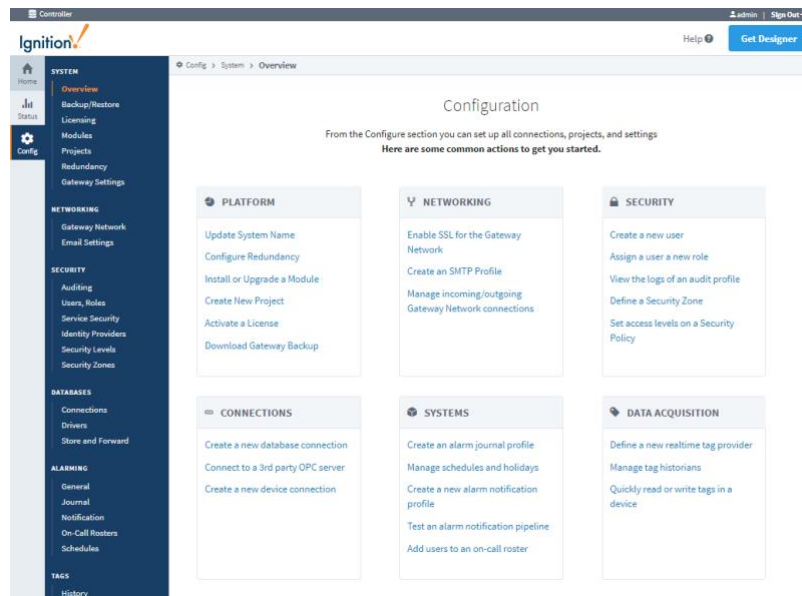


*Ilustración 6: Bloque Status del Gateway*

### Bloque 3: Configuration

En la página *Configuration* se pueden configurar todas las conexiones y proyectos, así como realizar todas las operaciones de *Gateway* y del sistema.

En la ilustración 7 se muestra el aspecto del bloque *Configuration*.



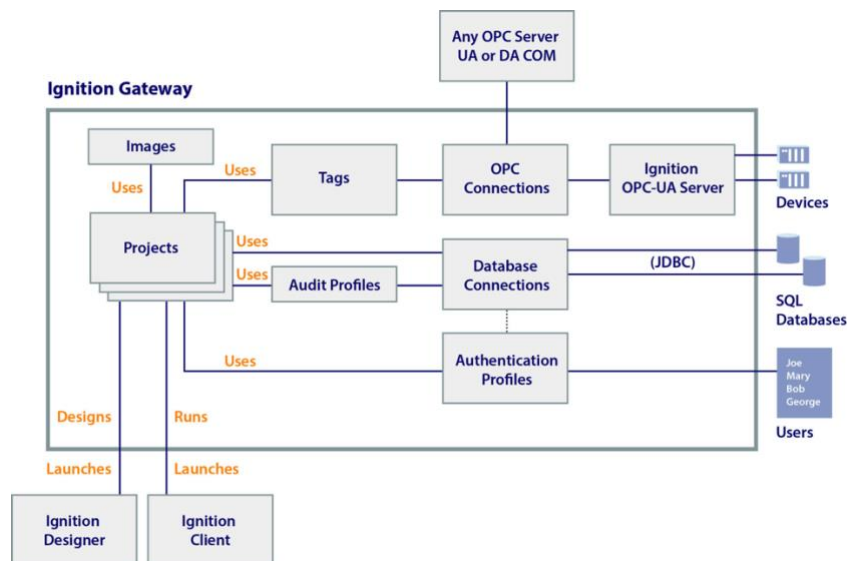
*Ilustración 7: Bloque Configuration del Gateway*

### Arquitectura del Gateway

La arquitectura del *Gateway* contiene muchas partes, cada una de las cuales le permite realizar una tarea específica. Las partes de *Gateway* son las siguientes (Inductive Automation, 2020):

- Gestión del sistema
- Gestión de proyectos y módulos
- Configuración de seguridad y auditoría
- Conectividad de base de datos
- Configuraciones de alarma y modificación
- Configuración del proveedor de etiquetas
- Conectividad OPC
- Configuración de administración empresarial

En la ilustración 8 se muestran las diferentes partes del *Gateway* y como funcionan los proyectos dentro del *Gateway*.



**Ilustración 8:** Arquitectura del Gateway  
Fuente: (Inductive Automation, 2020)

### 2.2.3 Módulos

Como se ha comentado anteriormente, *Ignition* posee una arquitectura modular. Los módulos de *Ignition* son archivos adicionales que agregan características y funcionalidades al sistema.

Los módulos de *Ignition* se dividen en cuatro tipos (Inductive Automation, 2020):

- Módulos Principales
- Módulos Adicionales
- Módulos de terceros
- Módulos creados por el usuario

En los siguientes puntos se describe cada uno de los tipos de módulos.

#### Módulos principales

Los módulos principales son módulos de uso común incluidos en una instalación típica. La mayoría de los sistemas tienen instalado al menos uno de estos módulos. Estos módulos son la base sobre la cual se construyen los proyectos.

Entre los módulos principales se encuentran:

- **Perspective:** Este módulo sirve para crear aplicaciones para su uso en dispositivos móviles y en navegadores web. Dichas aplicaciones pueden dar una visión general sobre el sistema SCADA. *Perspective* es un nuevo módulo que implementó *Ignition* recientemente y está en crecimiento.
- **Vision:** Este módulo es el que se utiliza para crear las aplicaciones SCADA. Permite crear gráficos con datos en tiempo real, identificar el estado de los indicadores, crear alarmas, etc.
- **Symbol Factory:** consiste en un “Toolbox” que posee un sin número de gráficos y objetos que ayudan al momento de crear las pantallas de la aplicación.
- **Reporting:** Permite la creación y diseño de reportes dinámicos. Los reportes pueden ser creados en varios formatos como: PDF, HTML, CSV y RTF.
- **SQL Bridge:** Este módulo permite la comunicación con las bases de datos.
- **Tag Historian:** Sirve para almacenar los datos en la base de datos.
- **Alarm Notification:** Permite la configuración de acuerdo con las necesidades de la empresa de cómo, dónde y por qué se generaron las alarmas, mediante el envío de correo electrónico hacia un específico grupo de usuarios
- **OPC UA:** *Ignition* permite la comunicación de la plataforma con varios tipos de dispositivos mediante el protocolo de comunicación OPC UA.

### Módulos Adicionales

Los módulos adicionales proporcionan a *Ignition* una funcionalidad adicional.

### Módulos de terceros

Además de los módulos proporcionados por *Inductive Automation*, otras compañías han desarrollado sus propios módulos utilizando el Kit de desarrollo de software de módulos. Un módulo de terceros útil es *Kymera Map Panel*, proporciona la capacidad de mostrar e interactuar con mapas de una serie de proveedores de mapas basados en *OpenStreetMap*. Permite crear aplicaciones de mapas dinámicas dentro de *Ignition* con puntos de interés, rutas y formas dibujadas en 2D. Este módulo no posee una versión de prueba por lo que no se han hecho pruebas con él en el presente proyecto.

El módulo de *Kymera Map* permite insertar mapas como el que se muestra en la ilustración 9.



**Ilustración 9:** Un mapa creado con el módulo de *Kymera Map Panel*  
Fuente: (Kymera systems, 2020)

#### 2.2.4 Arquitectura

*Ignition* se acopla a las necesidades de la infraestructura de la empresa. Por lo cual existen varios tipos de arquitecturas y conexiones que se pueden formar con los servidores de la plataforma. Se puede anclar el servidor en un solo campo, en múltiples campos o simplemente en alguna dirección de red en la nube para poder acceder remotamente a el.

Las arquitecturas en las que se puede implementar *Ignition* se resumen en la tabla 1 (Inductive Automation, 2020):

Arquitecturas	Descripción
<b>Arquitectura básica</b>	Es una arquitectura escalable con gestión central. <i>Ignition</i> funciona como un servidor central que sirve datos a todos los Clientes en la red. Las conexiones a PLCs, Bases de Datos y Clientes no tienen límite.
<b>Arquitectura redundante</b>	El <i>Ignition Gateways</i> tiene la capacidad de ser configurado en redundancia, cosa que permite una alta disponibilidad de datos hacia los clientes.

<b>Arquitectura store-and-forward</b>	<p>En muchas arquitecturas, las conexiones entre <i>Ignition</i>, las Bases de Datos y los PLCs remotos utilizan medios de comunicación inestables (GPRS, Satélite, WiMax, Radio, etc).</p> <p><i>Ignition</i> permite una configuración store-and-forward que asegura que no hay pérdida de datos remotos</p>
<b>Arquitectura Distribuida</b>	<p>La instalación de <i>Gateways</i> independientes en cada planta permiten la funcionalidad de SCADA para control local e historización. Una red corporativa permite compartir datos y pantallas del SCADA entre usuarios de las diferentes plantas.</p>
<b>Arquitectura de red</b>	<p><i>Ignition</i> permite el flujo de información entre la red de planta (con los PLCs) y la red de gestión (con las Bases de Datos).</p>
<b>Arquitectura para Control de Emergencia</b>	<p><i>Ignition</i> está preparado para actuar cuando una parte de su red queda desconectada del Gateway. Esto permite actuar sobre los sistemas de control dando seguridad a los procesos.</p>

**Tabla 1: Arquitecturas de Ignition**

### 2.2.5 Comparativa entre *Ignition* y otras plataformas

En la tabla 2 se hace una comparación entre *Ignition* y otras plataformas a nivel de bases de datos compatibles, sistemas operativos soportados y a nivel de licencias:

<b>Plataforma</b>	<b>Bases de datos compatibles</b>	<b>Sistemas operativos compatibles</b>	<b>Otras características</b>	<b>Licencias</b>
<i>Ignition de Inductive Automation</i>	<i>MySQL</i> <i>Microsoft SQL</i> <i>Oracle</i> <i>PostgreSQL</i>	Multiplataforma, no depende del sistema operativo Servidor basado en la web.	Curso de formación gratuito con certificación.  <i>Scripting en Python</i>	Una licencia por servidor.  Clientes ilimitados Desarrolladores ilimitados Tags ilimitados.

	<i>Firebird</i>			
	<i>IBM DB2</i>			
<i>Intouch de Wonderware</i>	<i>Microsoft SQL</i>	<i>Windows</i>	Mucho tiempo en el mercado <i>Scripting .Net</i>	Licencias basadas en número de <i>tags</i> .
<i>WinCC de Siemens</i>	<i>Microsoft SQL</i>	<i>Windows</i>	<i>Scripting VBscript</i>	Licencia basada en número de <i>tags</i>

**Tabla 2:** Comparativa Ignition y otras plataformas

## 2.3 Criterios y requisitos de diseño de la aplicación SCADA

Para el diseño de la interfaz SCADA objeto del presente proyecto se va a tener en cuenta varios criterios y normas, como la norma ISA SP101, HMI de alto rendimiento.

### 2.3.1 Criterios

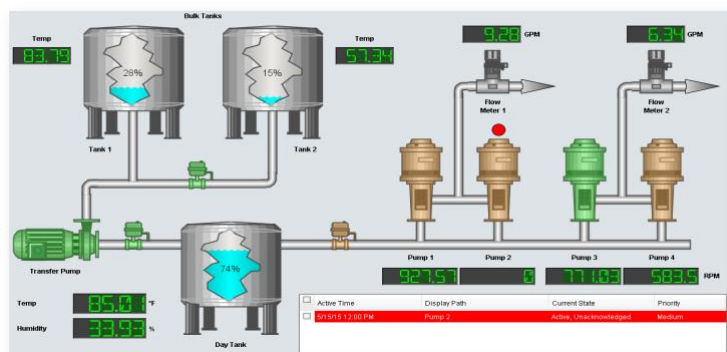
En general los criterios que se van a tener en cuenta para el diseño son los siguientes:

- Usar símbolos y pantallas estandarizadas.
- Procurar simetría, alineación y reparto uniforme de la información.
- Todo lo que signifique lo mismo o funcione igual debe representarse de la misma manera.
- Incrementar la información que se muestre en pantalla, eliminando todo lo innecesario para reconocer el estado actual del sistema.
- Diseñar gráficos de visión general y orientados a operaciones específicas.
- Utilizar elementos gráficos y fuentes de letras adecuadas.
- Indicar el estado actual del proceso de una manera intuitiva.
- La interfaz debe permitir la toma rápida de decisiones y realizar acciones de control sobre el sistema.
- Los colores deben ser representativos contribuyendo al significado y relevancia del estado actual de los componentes del sistema.

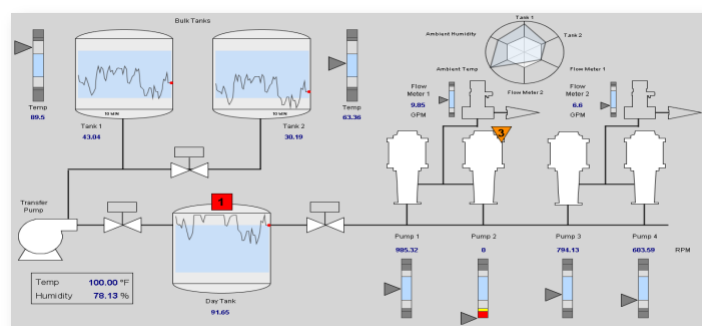
### 2.3.2 HMI alto rendimiento

El diseño de HMI esta avanzando cada vez mas hacia el diseño de alto rendimiento. El método de diseño de HMI de alto rendimiento ayuda a garantizar una interacción rápida y eficaz entre el usuario y el sistema. El diseño de HMI de alto rendimiento se basa en llamar la atención únicamente sobre los indicadores mas necesarios o críticos de la interfaz. Esta técnica de diseño ayuda al usuario a ver y responder a las anomalías del sistema de manera eficiente, así como tomar decisiones de forma rápida y eficaz. Las HMI de alto rendimiento utilizan indicadores básicos y simples, por lo general, usan colores en escala de grises en lugar de los gráficos tradicionales. Conceptualmente, el HMI de alto rendimiento contrasta visualmente los estado críticos y no críticos, con el objetivo de guiar rápidamente al usuario a la fuente de alguna anomalía en el sistema.

En la ilustración 10 se muestra una comparación entre un HMI tradicional y un HMI de alto rendimiento.



a)



b)

**Ilustración 10:** HMI tradicional (a) y HMI alto rendimiento (b)  
Fuente: (Inductive Automation, 2020)



La biblioteca de componentes de *Ignition* consta de varios componentes de alto rendimiento, además, la posibilidad de importar iconos en formato SVG facilita mucho la tarea de diseñar HMI de alto rendimiento.

### **2.3.3 La norma ISA101**

La norma ISA 101 HMI pretende marcar una serie de convenciones y normas a la hora de diseñar interfaces HMI utilizados en la automatización industrial de máquinas y procesos.

En la creación de una interfaz HMI intervienen varias ciencias, como la filosofía, psicología, lingüística, etc. Que aportan de gran forma al desempeño de cada uno de los componentes de la interfaz. Con los visualizadores se pueden representar las características principales del sistema como son valores medidos, variables de control, mensajes de ayuda para el operador e incluso alarmas del sistema.

El operador debe ser capaz de obtener la mayor cantidad de información en el menor tiempo posible, por medio de las herramientas visuales que la interfaz le proporciona.

El operador va a estar manejando la interfaz alrededor de 8 horas diarias, por lo que es necesario tomar en cuenta los factores humanos, ya que dependiendo del uso que le dé el usuario a la interfaz depende el éxito del sistema. Los factores más importantes que se deben tomar en cuenta son la percepción visual, las formas de pensamiento y el comportamiento del usuario.

Para cumplir con los factores que se establecieron, el HMI deberá cumplir con las siguientes características:

- Visualizadores de texto y gráficos, para mejorar el desempeño y la gestión de la información del operador.
- Sistema que aseguran una simple y sencilla interacción con la aplicación.

### **2.3.4 Requisitos**

Los requisitos de Suez para diseñar la aplicación en *Ignition* se recogen en los siguientes puntos:

- Identificar los diferentes módulos, plantillas, funcionalidad, etc., de la aplicación *Wonderware*.
- Realizar un análisis y descripción técnica de alto nivel de cómo cada una de las partes descompuesta de *Wonderware* puede ser cubierta con la plataforma *Ignition*.

En caso de que alguna de las partes descompuesta no sea cubierta, se describirá una aproximación que pueda suplir la función correspondiente.

Para llevar a cabo los requisitos anteriores, se estudia la aplicación diseñada en *Wonderware* de un municipio de la zona de Alicante, para extraer toda la información posible necesaria para diseñar la aplicación SCADA en *Ignition*.

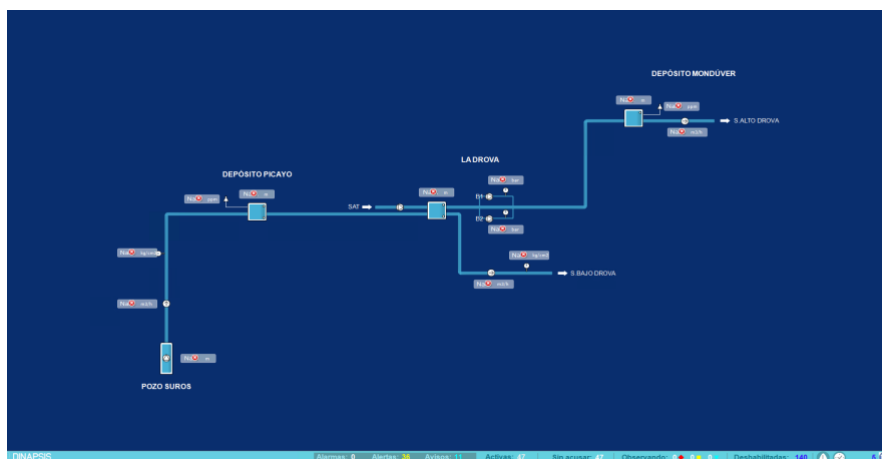
#### 2.3.4.1 Análisis de la plataforma Suez

El municipio cuenta con un mapa general de abastecimiento que se compone de las siguientes cuatro estaciones:

- Estación La Drova.
- Estación Pozo Suros.
- Estación Depósito Picayo.
- Estación Depósito Monduver.

Los módulos de la aplicación *Wonderware* analizados se recogen en los siguientes puntos:

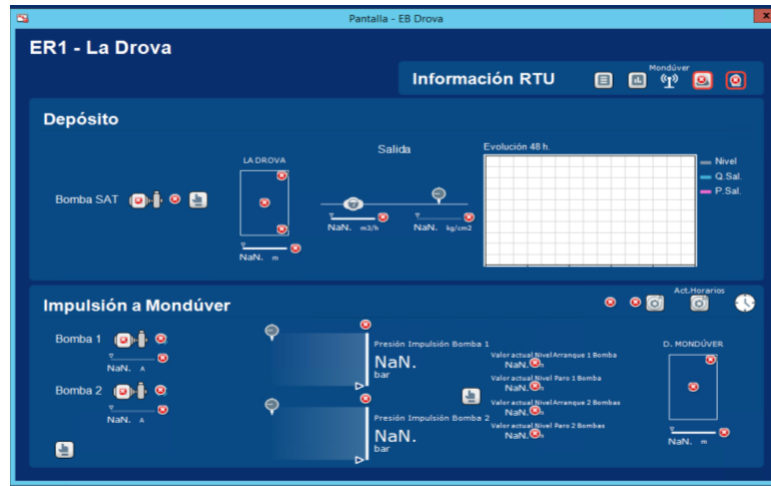
- a- **Estructura y navegación:** Tras analizar la estructura y la navegación de la aplicación diseñada en *Wonderware*, se observa que todas las estaciones se representan en forma de un sinóptico, este sinóptico aparece como una ventana principal de la aplicación. en la ilustración 11 se muestra un ejemplo de un sinóptico.



*Ilustración 11: Ejemplo de un sinóptico*

Desde el sinóptico se puede acceder a las siguientes ventanas:

- 1- **Ventana de estación:** La ventana de estación aparecerá como una ventana de tipo *Popup*. Muestra toda la información relacionada con la estación. En la ilustración 12 se señala un ejemplo de una ventana de estación.



*Ilustración 12: Ejemplo de una ventana de estación*

- 2- **Faceplates:** Los *faceplates* son ventanas de tipo *Popup*, sirven para controlar y obtener información sobre los distintos elementos que componen el sinóptico. En la ilustración 13 se ilustra un *faceplate* de una bomba.



*Ilustración 13: Faceplate de una bomba*

Respecto a la navegación, la aplicación estudiada contiene una barra superior fija, en la cual se puede navegar por los diferentes sinópticos y ver información de usuario. Ver la ilustración 14.



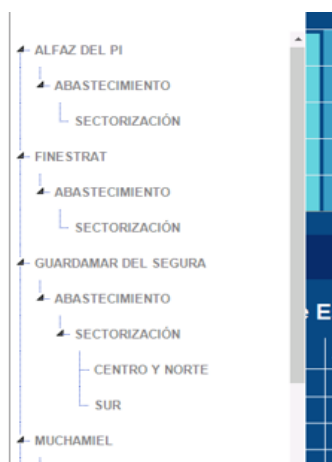
*Ilustración 14: Barra superior de la aplicación Wonderware*

La aplicación analizada contiene también una barra lateral que dispone de un árbol de navegación de los sinópticos general y un buscador de estaciones. La barra lateral se indica en la ilustración 15.



*Ilustración 15: Barra lateral de la aplicación Wonderware*

- **Árbol de Sinópticos General:** Se muestra los niveles jerárquicos de los sinópticos generales/cuadros de mando, pudiendo clicar en cada uno de ellos para acceder al sinóptico asociado. Ver ilustración 16.



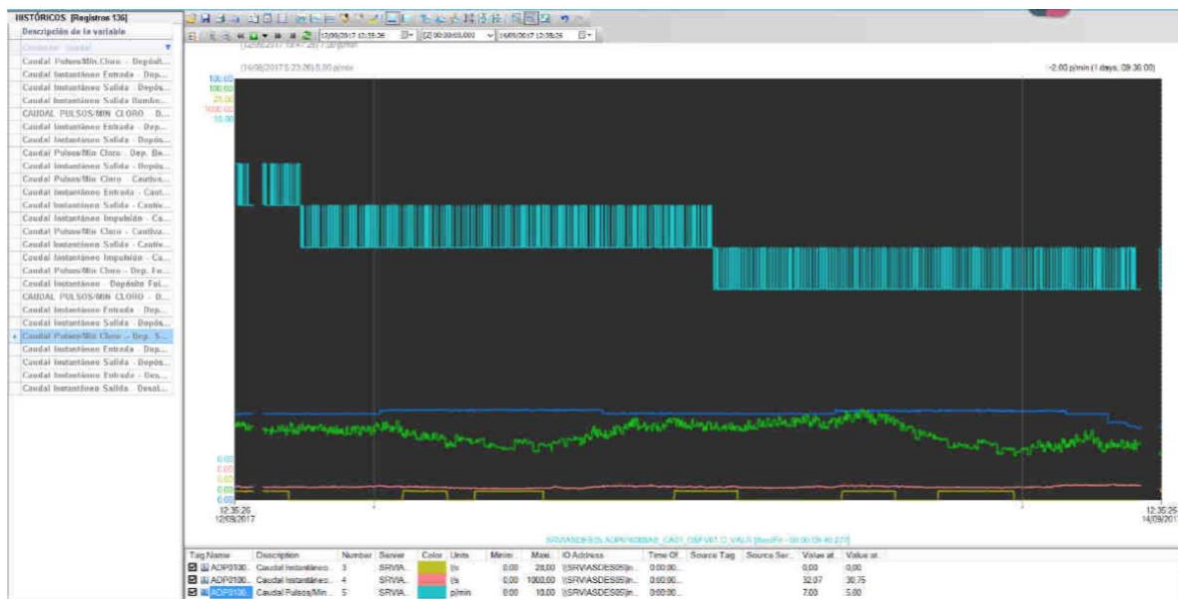
*Ilustración 16: Árbol de Sinópticos General*

- **Buscador de Estaciones:** Se muestran los resultados de las estaciones cuyo nombre coincida con el indicado en cuadro de texto del buscador. Una vez listadas las estaciones, se puede seleccionar la estación en concreto que se busca. Tras pulsar sobre la estación que se desea, se cambia dicho listado por el de los sinópticos donde aparece información sobre la estación. Al pinchar sobre el sinóptico deseado, se presenta el mismo en el área de explotación de datos. Los sinópticos, estarán tipificados como se muestra en la ilustración 17.



*Ilustración 17: Buscador de estaciones*

- b- **Gestión de históricos:** El módulo de históricos de la aplicación analizada disponer de varias opciones de visualización de datos históricos:
- **Históricos por estación:** desde la ventana de cada estación, se puede acceder a una ventana de históricos en la que se representan todas las variables historizadas de la estación.
  - **Histórico por elemento:** Desde los sinópticos o las ventanas de estaciones se seleccionan las variables de los elementos que se quiera estudiar.
  - **Históricos genéricos:** desde la barra superior descrita anteriormente se puede acceder a una ventana de históricos. Dicha ventana contiene una lista de todas las señales historizadas. La ventana de históricos genéricos se representa en la ilustración 18.



**Ilustración 18:** Ventana de históricos genéricos de la aplicación Wonderware

- c- **Gestión de alarmas:** En la aplicación analizada, existe una ventana de gestión de alarmas donde se muestra en forma de tabla alarmas en tiempo real y también un histórico de alarmas.

La ventana de gestión de alarmas se señala en la ilustración 19.

Control de alarmas actuales									
<div> <div>TODAS</div> <div>ALARMAS</div> <div>ALERTAS</div> <div>AVISOS</div> <div>OBSERVADO</div> <div>Enviado</div> </div> <div> <div>Mostrar de alarmas:</div> <div>Rutinas</div> <div>seleccione</div> <div>Cambiar</div> <div>Propiedades de alarmas</div> <div>Quitar alarmas</div> <div>Actualizar</div> <div>Actualizar</div> <div>Actualizar</div> </div>									
Gravedad	Estado	Tagname	Descripción		Tipo	Fecha	Limite	Valor Actual	
1	UNACK	RTN	GDS from GR_SISCON	Last alarm communication to GDS.	Comm	23.08.2017 09:42:11			
1	UNACK	RTN	GDS01 from GR_SISCON	Last alarm communication to GDS01.	Comm	23.08.2017 09:42:11			
1	UNACK	RTN	GDS01_01 from GR_SISCON	Last alarm communication to GDS01_01.	Comm	23.08.2017 09:42:11			
1	UNACK	RTN	GDS01_ABSC from GR_SISCON	Last alarm communication to GDS01_ABSC.	Comm	23.08.2017 09:42:11			
1	UNACK	RTN	TRV01_ABD1 from GR_SISCON	Last alarm communication to TRV01_ABD1.	Comm	23.08.2017 09:42:11			
1	UNACK	RTN	TRV01_ABD1_ST01 from GR_SISCON	Last alarm communication to TRV01_ABD1_ST01.	Comm	23.08.2017 09:42:11			
1	UNACK	RTN	TRV01_ABD1_ST01_D001 from GR_SISCON	Last alarm communication to TRV01_ABD1_ST01_D001.	Comm	23.08.2017 09:42:11			
1	UNACK	RTN	TRV01_ABD1_ST01_SP01 from GR_SISCON	Last alarm communication to TRV01_ABD1_ST01_SP01.	Comm	23.08.2017 09:42:11			
1	UNACK	RTN	TRV01_ABTA from GR_SISCON	Last alarm communication to TRV01_ABTA.	Comm	23.08.2017 09:42:11			
1	UNACK	RTN	TRV01_ABTA_CUB1 from GR_SISCON	Last alarm communication to TRV01_ABTA_CUB1.	Comm	23.08.2017 09:42:11			
3	UNACK	GDS01_pruebas.Alertas	Enter attribute description		DSC	29.05.2017 12:44:20	true	true	
2	UNACK	GDS010001AB_0P01_IVB0A01_V_NIVE.HI	Nivel Depósito Mancaya		HI	25.05.2017 18:22:44	150.0	NaN	
2	UNACK	GDS010003AB_GN01_PRB0A01_V_PRES.HI.HI	Presión Meteorológica		HI.HI	18.05.2017 16:44:44	2.0	NaN	
2	UNACK	GDS010003AB_GN01_PRB0A01_V_PRES.HI	Presión Meteorológica		HI	18.05.2017 16:44:44	1.5	NaN	
3	UNACK	GDS01_pruebas.Analogica1.o	Enter attribute description		Lo	18.05.2017 14:42:44	25.0	0	
3	UNACK	GDS01_pruebas.Analogica1.o	Enter attribute description		Lo	18.05.2017 14:42:44	10.0	0	
3	UNACK	GDS01_pruebas.Analogica2.o	Enter attribute description		Lo	18.05.2017 14:42:44	25.0	0	
3	UNACK	GDS01_pruebas.Analogica2.o	Enter attribute description		Lo	18.05.2017 14:42:44	10.0	0	
3	UNACK	GDS01_pruebas.Analogica3.o	Enter attribute description		Lo	18.05.2017 14:42:44	25.0	0	
3	UNACK	GDS01_pruebas.Analogica3.o	Enter attribute description		Lo	18.05.2017 14:42:44	10.0	0	
3	UNACK	GDS01_pruebas.Analogica4.o	Enter attribute description		Lo	18.05.2017 14:42:44	25.0	0	
3	UNACK	GDS01_pruebas.Analogica4.o	Enter attribute description		Lo	18.05.2017 14:42:44	10.0	0	

**Ilustración 19:** Ventana de gestión de alarmas de la aplicación Wonderware

Además, en los sinópticos se visualiza cuando ocurre una alarma mediante cambio de color en elemento en el que se produce la alarma.

En la ilustración 20 se muestra como se comporta el elemento de bomba cuando ocurre una alarma.



*Ilustración 20: Bomba en Alarma*

**d- Otros módulos analizados:**

- **Aplicación basada en objetos:** Un objeto puede definirse como un elemento que se programa solo una vez y se podrá utilizar en varias partes de la aplicación. contiene tanto la parte grafica de un elemento como la parte de datos (señales). Los objetos instanciados de una plantilla se caracterizan por la propiedad de herencia, es decir un objeto puede heredar las características de otro.
- **Notificación de alarmas:** La notificación de alarmas se realiza mediante un Software externo llamado *Nuntius*. Se trata de un cliente OPC UA que recibe los datos de alarmas, y según estos datos se notifica a los usuarios correspondientes.
- **Trazador de variables:** La ventana de estación contiene un icono que permite acceder a una tabla en la cual se recogen todas las variables de la estación, en la tabla se indica las siguientes informaciones:
  - Valor en tiempo real.
  - La fecha/hora del último dato recibido.
  - Descripción, unidades, rango, etc.

El objetivo del trazador de variables es la facilidad de encontrar errores de una variable

- **Seguridad:** La seguridad de acceso a la aplicación analizada se basa en el directorio activo, además la seguridad de acceso a la información depende del rol o el ámbito del usuario que se accede a la aplicación.
- **Generador de instancias:** El objetivo del generador de instancias es facilitar la creación de instancias. En las migraciones, automatizar en la medida de lo posible, la asociación de las antiguas variables con las nuevas.



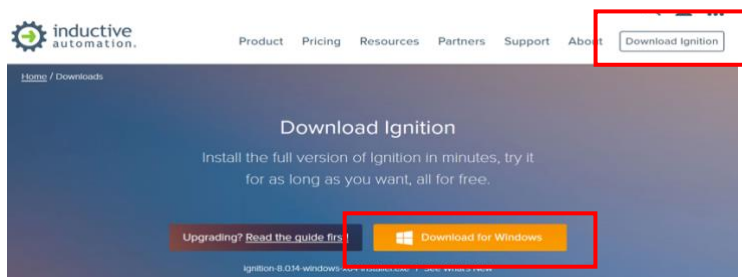


## 3 CAPÍTULO 3: IMPLEMENTACIÓN

En el presente capítulo se explicarán las tareas llevadas a cabo antes de diseñar la aplicación SCADA, como la descarga del Software *Ignition*, y la conexión a la base de datos. También se describirá todas las actividades realizadas para el diseño de la aplicación.

### 3.1 Instalación de software *Ignition*

El primer paso para el diseño de la aplicación SCADA es instalar el software *Ignition*. Para instalar *Ignition* es necesario tener instalada la plataforma de *Java*. *Ignition* se descarga de la página de *Inductive Automation* ([www.inductiveautomation.com](http://www.inductiveautomation.com)). Una vez se completa el registro en la misma página. Para la descarga se hace clic sobre el botón **Download Ignition**, y posteriormente se hace clic en el botón **Download for Windows**. Ver ilustraciones 21 y 22.



**Ilustración 21:** Página de descarga de *Ignition*

**Ilustración 22:** Página de Registro en la página de Inductive Automation

Una vez descargado y instalado, se inicia la página del Gateway. Cuando se inicia la página del Gateway por primera vez, se debe aceptar la licencia, introducir un usuario y contraseña, y por último especificar el puerto. el puerto por defecto es el 8088.

### 3.2 Conexión con la base de datos

Una vez instalado *Ignition*, el siguiente paso es investigar que base de datos se va a utilizar para el proyecto. Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacene de forma electrónica en un sistema informático (Oracle, 2020).

Para el presente proyecto, la base de datos es necesaria para el registro de datos históricos y para el registro de alarmas.

Las bases de datos compatibles con *Ignition* son (Inductive Automation, 2020):

- *MySQL*
- *Microsoft SQL Server*
- *Oracle*
- *PostgreSQL*
- *Firebird*
- *IBM DB2*

Tras estudiar todas las posibles bases de datos que pueden usar en *Ignition* y compararlas, se elige la base de datos *PostgreSQL* debido a las siguientes ventajas:

- Código abierto y gratuito: *PostgreSQL* es un gestor de base de datos de código libre y completamente gratuito.
- Multiplataforma: *PostgreSQL* es compatible con prácticamente todas las tecnologías y sistemas operativos actuales.
- Gran volumen de datos.
- Es la mas utilizada por los usuarios de *Ignition*.

Una vez instalado el gestor de la base de datos de *PostgreSQL*, el siguiente paso es realizar la conexión con *Ignition*. Como ya se ha mencionado en el apartado de Gateway page, todas las conexiones de *Ignition* se realizan a través de la página del Gateway.

Para conectar la base de datos, se dirige al bloque *Configuration*, luego a *Databases > Connections*. (ver la ilustración 23).



**Ilustración 23:** Conexión a una base de datos

En la página de *Connections*, se controlan todas las bases de datos que están conectadas con *Ignition*, se puede ver el estado en la que se encuentran las conexiones ya existentes, así como editarlas o borrarlas. Para crear una nueva conexión se pulsa sobre *Create new Database Connection*. ver la ilustración 24.

Name	Description	JDBC Driver	Translator	Status	
BaseDatos2		PostgreSQL	POSTGRES	Valid	<a href="#">delete</a> <a href="#">edit</a>
BaseDeDatos		MySQL	MYSQL	Valid	<a href="#">delete</a> <a href="#">edit</a>

[→ Create new Database Connection...](#)

**Note:** For details about a connection's status, see the [Database Connection Status](#) page.

**Ilustración 24:** Crear una nueva conexión a la base de datos

El siguiente paso es elegir *PostgreSQL* (ver la ilustración 25), y rellenar los datos de conexión que se muestran en la tabla 3.

☐ **MySQL**  
The official MySQL JDBC Driver, Connector/J.

☐ **Oracle Database**  
The Oracle Database JDBC driver.

☒ **PostgreSQL**  
The official PostgreSQL JDBC Driver.

[Next >](#)

**Ilustración 25:** Seleccionar la base de datos

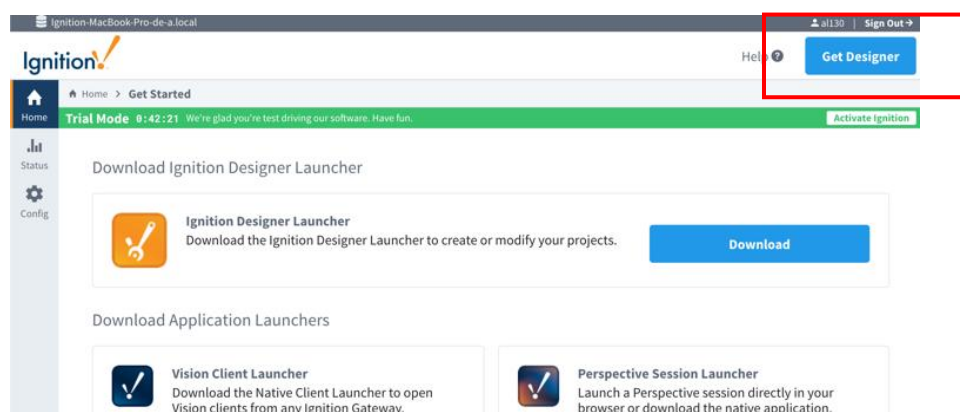
<b>Name</b>	BaseDatos2
<b>Description</b>	Base de datos predeterminada
<b>JDBC Driver</b>	PostgreSQL
<b>Connect URL</b>	jdbc:postgresql://localhost:5432/Ignition
<b>Username</b>	postgres
<b>Password</b>	postgres
<b>Password</b>	postgres

*Tabla 3: Datos de conexión a la base de datos*

### 3.3 Creación de un proyecto

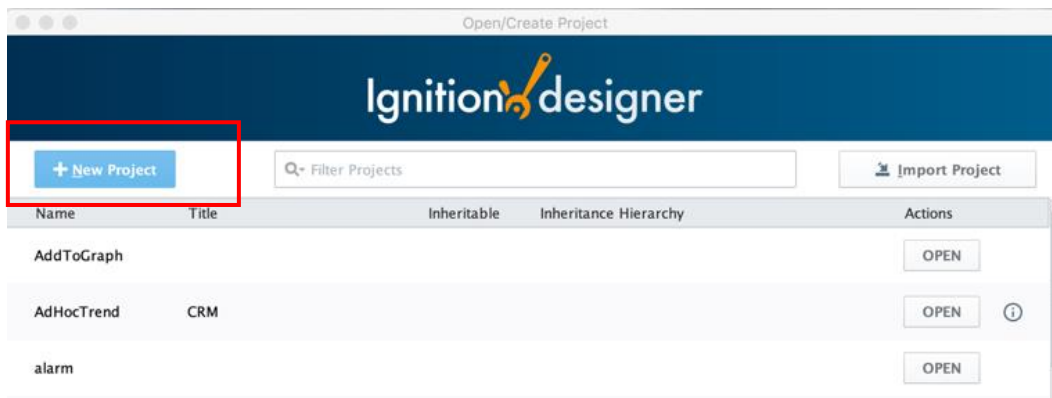
Para crear un nuevo proyecto y empezar a diseñar la aplicación SCADA, es necesario descargar “*Designer*”. Se trata del programa que se utiliza para realizar la configuración y el diseño de las aplicaciones SCADA.

Para descargar “*Designer*” se accede a la página del *Gateway*, y se descarga haciendo clic en botón *Get Designer*. Ver la ilustración 26.



*Ilustración 26: Descarga de "El diseñador"*

Una vez descargado y instalado “*Designer*”, se ejecuta desde el escritorio para crear un nuevo proyecto. Una vez ejecutado se ingresa el usuario y la contraseña del usuario, se hace clic sobre el botón *New Project* luego se rellenan los datos del nuevo proyecto. ver ilustración 27 y 28.



*Ilustración 27: Crear un nuevo proyecto*

*Ilustración 28: Datos de un proyecto nuevo*

### 3.4 Creación de ventanas

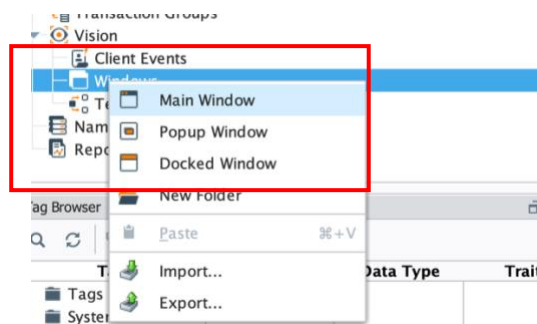
Una vez creado el proyecto, se procede a crear las ventanas que van a componer la aplicación. Como ya se ha mencionado en el apartado de módulos de *Ignition*, el módulo que se utiliza para crear las aplicaciones SCADA es *Vision*. En el módulo de *Vision* se pueden crear tres tipos de ventanas: Ventanas principales, ventanas emergentes y ventanas acopladas.

A continuación, se describe cada uno de los tipos de ventanas.

- **Una ventana principal:** es aquella que está configurada para ser maximizada al abrirse. En este proyecto las ventanas principales son los sinópticos y los mapas de proceso.
- **Una ventana emergente:** es una ventana cuya que se abre como flotante y no está maximizada. Se utilizan las ventanas emergentes para crear *faceplates* de los diferentes componentes del sistema, así como para crear ventanas de estaciones, ventana de gestión de alarmas, ventana de gestión de históricos, etc.

- Una **ventana acoplada** es una ventana fija en una posición predeterminada, y siempre va a estar mostrada. Las ventanas fijas que posee la aplicación son dos, una ventana fija superior, y otra ventana fija inferior, las dos ventajas se describirán en los siguientes apartados.

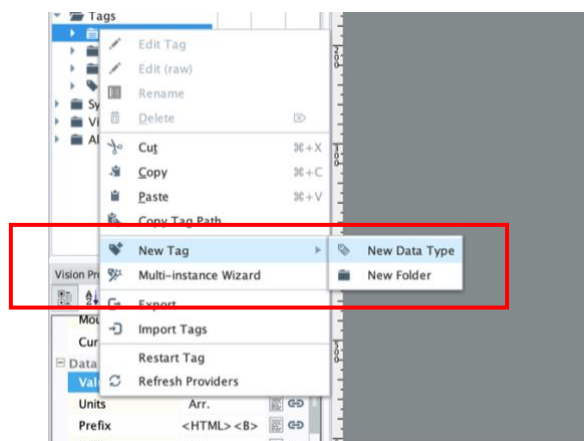
Para crear una nueva ventana, se dirige a **Vision** que se encuentra en la ventana **Project Browser**, luego se hace clic con el botón derecho en la carpeta que **Windows** y se elige el tipo de ventana a crear. Ver ilustración 29.



*Ilustración 29: Crear una nueva ventana*

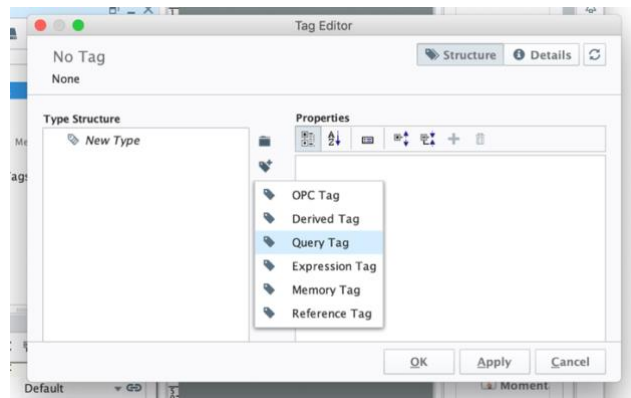
### 3.5 Creación de UDTs

Los UDTs (tipos definidos por el usuario) ofrecen la capacidad de aprovechar los principios de diseño de datos orientados a objetos. Al crear todos los UDT necesarios para el proyecto, se puede generar instancias de etiquetas de UDT. Todas las instancias comparten todas las propiedades, así se ahorra mucho tiempo y trabajo a la hora de realizar una modificación. Para crear un nuevo UDT, se dirige a la ventana **Tag Browser**, luego se hace clic con el botón derecho en la carpeta **Data Types** y seleccionar **New Tag > New Data Type**. Tal y como se indica en la ilustración 30.



*Ilustración 30: Crear un nuevo UDT*

Para añadir *Tags* al UDT, se hace clic sobre el botón Agregar *Tags* y se elige el tipo del *Tag* a añadir (Ver ilustración 31), Para el presente proyecto todos los *tags* creados son de tipo memoria.

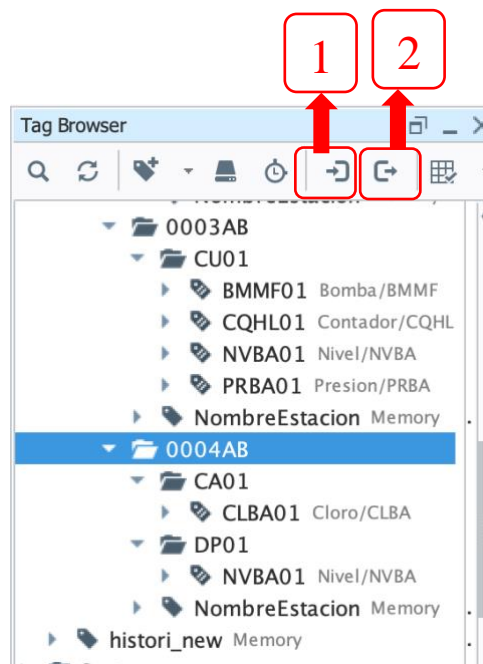


**Ilustración 31:** Agregar un Tag al UDT

*Ignition* ofrece la posibilidad de exportar e importar UDTs o *Tags* en formato XML o CSV. Esto permite realizar cambios en la estructura de UDTs fuera de *Ignition*. La función de importar/exportar UDTs es importante cuando se desea migrar a *Ignition* un SCADA desarrollado a en otra plataforma

Para exportar e importar UDTs se dirige a la ventana **Tag Browser**, se selecciona el UDT que se quiere exportar/importar y se pulsa sobre uno de los botones que se indican en la ilustración 32.

El botón 1 es para importar y el botón 2 para exportar.

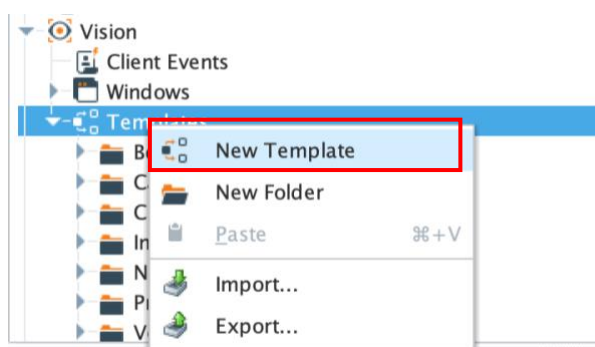


**Ilustración 32:** Exportar/ importar UDT

### 3.6 Creación de *Templates*

Los *templates* o las plantillas son el elemento gráfico de los *datatypes*, ya que cada uno de los *datatypes* creados se asocia con una plantilla gráfica. Las plantillas se crean solo una vez, luego las instancias de un *datatype* tendrán los mismos componentes y propiedades de la plantilla, y se actualizarán automáticamente a medida que se realicen cambios en la plantilla.

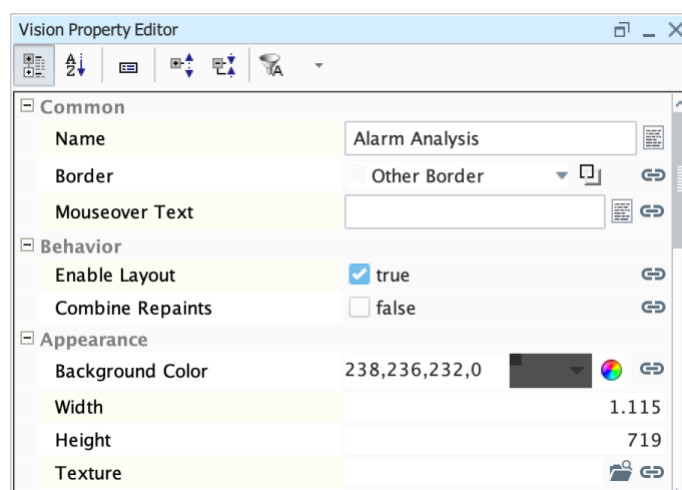
Para crear una plantilla, hay que dirigirse a ***Vision > Templates > New Template***. Tal y como se puede ver la ilustración 33.



***Ilustración 33: Crear un nuevo Template***

Una vez creado un *template* se despliega la ventana de trabajo donde se puede agregar los objetos que se necesitan para el *template* (componentes, textos, etc).

Para configurar las propiedades del *template* se dirige a la ventana “***Property Editor***”. Dicha ventana se muestra en la ilustración 34.

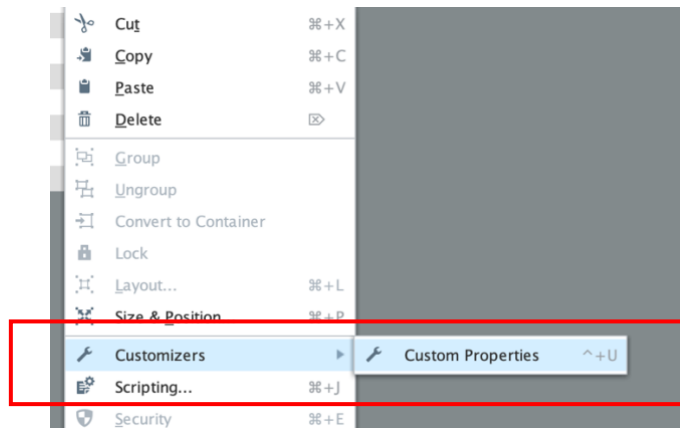


***Ilustración 34: Property Editor de un Template***

Los *templates* pueden ser personalizados de acuerdo con las necesidades del usuario. Para ello existen propiedades personalizadas que se pueden crear al *template*, dentro de estas propiedades existe la posibilidad de asociar el *template* creado con un determinado *datatype*.

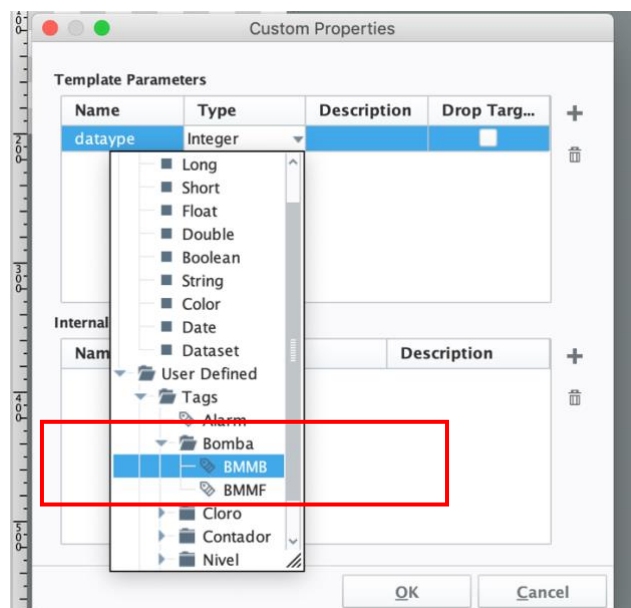


Para ello, se hace clic con el botón derecho sobre la zona de trabajo del *template* y se dirige a **Customizers > Custom Properties**. Ver ilustración 35.



*Ilustración 35: Acceso a Custom Properties de un template*

Posteriormente, en la ventana que aparece a continuación, en **Template Parameters > Type** se selecciona el *datatype* que se quiere asociar con el *template*. Ver ilustración 36.



*Ilustración 36: Custom Properties de un template*

Los *templates* asociados con *datatypes* cumplen la función de los objetos descritos en el apartado: 2.3.4.1 Análisis de la plataforma Suez.

### 3.7 Scripting en Ignition

Además de las propiedades que tienen los componentes que se utilizan en *Ignition*, existe la posibilidad de emplear los scripts para realizar alguna acción adicional o específica que no se encuentra dentro de dichas propiedades. *Ignition* utiliza el lenguaje de programación Python. Existen dos tipos de scripts, los que pueden ser ejecutados a nivel de Gateway o a nivel de cliente. Los scripts ejecutados a nivel de Gateway se ejecutan en el Gateway de la aplicación, mientras que los scripts ejecutados a nivel del cliente se ejecutan en cada uno de los clientes o usuarios que están utilizando la aplicación. El módulo de scripting en *Ignition* ha permitido diseñar de forma fácil y rápida las funciones de la aplicación desarrollada en *Wonderware* con el lenguaje de programación **.net**.

### 3.8 Aplicación SCADA

#### 3.8.1 UDTs Creados

Los UDTs creados en el presente proyecto se resumen en la tabla 4.

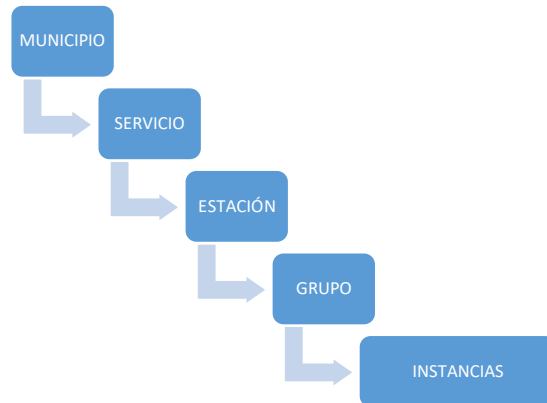
UDT	DESCRIPCIÓN	ETIQUETAS	DESCRIPCIÓN
BMMB	Bomba Básica	E_MARC	Estado de marcha
BMMF	Bomba Avanzada	A_CARR	Número de arranques
		A_CHOR	Números de horas de funcionamiento
		E_AVER	Estado avería
		E_FMAR	Estado fallo marcha
		E_FPAR	Estado fallo paro
		E_MARC	Estado de marcha
		T_AUTO	Telemando control automático
		T_MANU	Telemando control manual
		T_MARC	Telemando marcha
		T_PARO	Telemando paro
		V_INT	Valor intensidad
CLBA	Cloro	V_CLOR	Valor cloro
CQHL	Contador	V_CAUD	Valor caudal
NVBA	Nivel	V_NIVE	Valor nivel
PRBA	Presión	V_PRES	Valor presión

**Tabla 4:** UDTs Creados

### 3.8.2 InstanciasCreadas

Una vez creados los UDTs, el siguiente paso es crear las instancias. Para la creación de las instancias se ha definido la estructura que se expone en la ilustración 37.

**Municipio/Servicio/Estación/Grupo/**



*Ilustración 37: Estructura de carpetas del proyecto*

Para ahorrar tiempo y trabajo a la hora de creación de instancias, y para evitar errores de creación de instancias, se ha creado un generador de instancias en una ventana utilizando Scripting de *Ignition*. El generador de instancias tiene como objetivo crear instancias de forma automática, y tiene el aspecto que se representa en la ilustración 38.

Path:	
Municipio:	<input type="text"/>
Servicio:	AB <input type="button" value="v"/>
Estación:	<input type="text"/>
Grupo:	<input type="text"/>

Tipo de instancia:		Cantidad:
<input type="checkbox"/> Bomba	Seleccionar tipo <input type="button" value="v"/>	<input type="text" value="0"/>
<input type="checkbox"/> Cloro	Seleccionar tipo <input type="button" value="v"/>	<input type="text" value="0"/>
<input type="checkbox"/> Presión	Seleccionar tipo <input type="button" value="v"/>	<input type="text" value="0"/>
<input type="checkbox"/> Nivel	Seleccionar tipo <input type="button" value="v"/>	<input type="text" value="0"/>

*Ilustración 38: Generador de Instancias*

El generador de instancias consta de los siguientes elementos:

- 1- **Path:** es la ruta de la carpeta donde se insertarán las instancias. Hay que indicar el nombre del municipio, el servicio, la estación y el grupo. Al introducir el *Path*, si éste no existe se creará con la estructura mencionada en la ilustración 38.

- 2- **Tipo de instancia y cantidad:** el tipo de instancia se refiere al *Datatype*, si el *Datatype* tiene mas de un tipo, se selecciona el tipo en la lista, y en Cantidad se introduce la cantidad de instancias a crear.

El generado de instancias se ha programado utilizando *Scripting* de *Ignition*. Para consultar el código utilizado, referirse al Anexo A: Código del generador de instancias .

Las instancias creadas en el presente proyecto se resumen en la tabla 5.

ESTACION	GRUPO	UDT	CANTIDAD INSTANCIAS
0001	DP01	CLBA	1
		CQHL	1
		NVBA	1
		PRBA	1
	GR01	BMMF	1
	GR02	BMMF	2
		PRBA	2
0002	CA01	CLBA	1
	DP01	CLBA	1
		CQHL	1
		NVBA	1
0003	CU01	BMMF	1
		CQHL	1
		NVBA	1
		PRBA	1
0004	CA01	CLBA	1
	DP01	NVBA	1

**Tabla 5:** Tablas de las instancias del proyecto

### 3.8.3 Descripción da tags

La descripción de *tags* tiene como objetivo permitir al usuario identificar rápidamente los *tags* en la aplicación, por ejemplo, en la ventana de alarmas y en la ventana de históricos. En *Ignition* la descripción de un *tag* se introduce a la hora de crearlo. Para automatizar la introducción de *tags* en todo el proyecto, se ha programado un código utilizando *Scripting* de *Ignition*. Para consultar el código, referirse al Anexo B: Código de descripciones de *tags*.

### 3.8.4 Tabla de señales

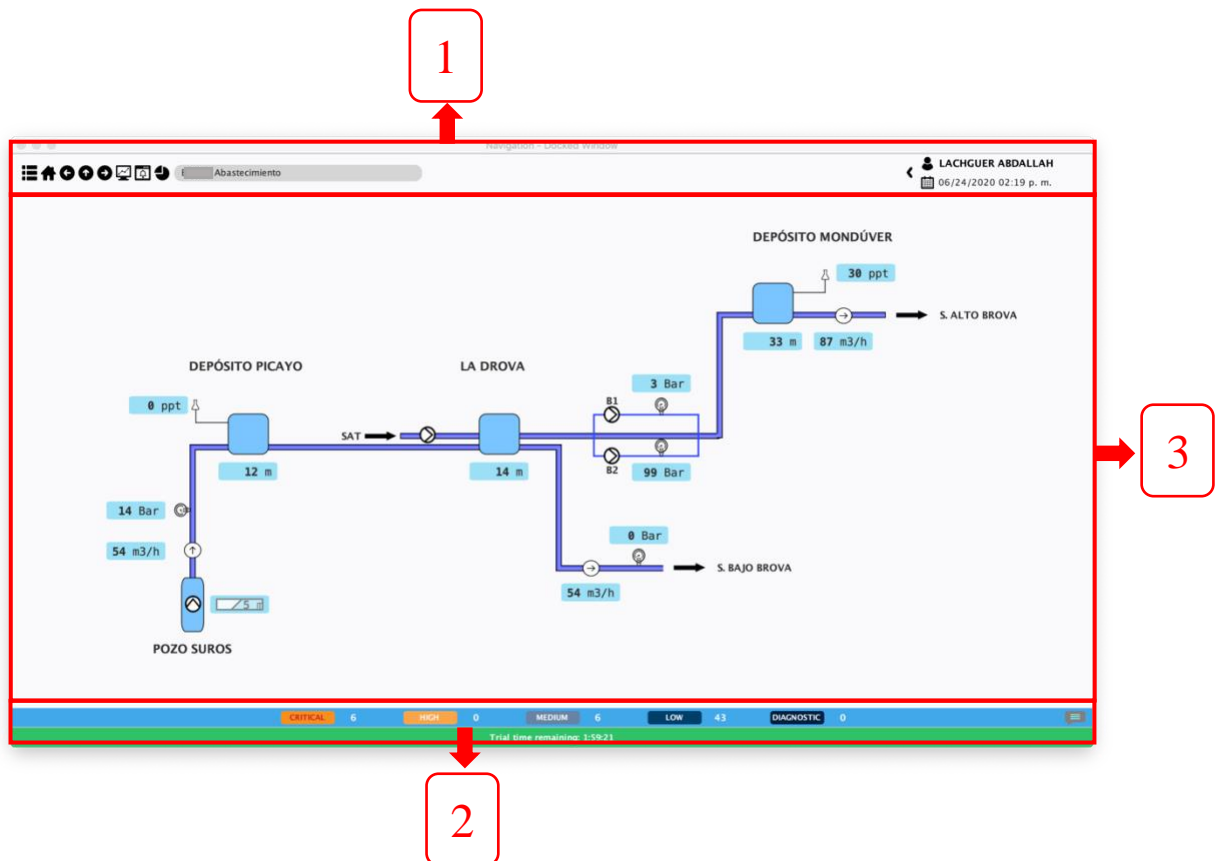
Para consultar la tabla de señales del proyecto, referirse al Anexo C: Lista de señales del proyecto.

### 3.8.5 *Templates* creados

Para consultar la tabla de *templates* creados en el proyecto, referirse Anexo D: Lista de *templates* del proyecto.

### 3.8.6 Interfaz de la aplicación

Cuando un usuario se conecta a la aplicación desarrollada, la interfaz de la misma tiene el aspecto que se muestra en la ilustración 39.

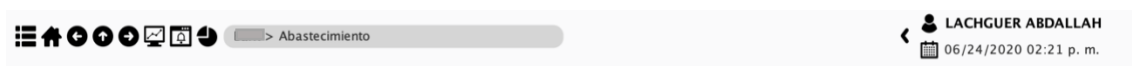


**Ilustración 39:** Interfaz de la aplicación desarrollada

La interfaz se compone de tres ventanas: una ventana superior, una ventana inferior y otra ventana principal. Las ventanas se describirán en los siguientes apartados.

#### 1- Ventana superior

La ventana superior es una ventana de tipo “*Docked*” con posición al norte de la pantalla. En la ilustración 40 se indica la ventana superior.







**Ilustración 40:** Ventana superior de la aplicación desarrollada

La ventana superior diseñada permite realizar las siguientes funciones:




- 1- Mostrar la ruta de la ventana principal activa.
- 2- Permite al usuario navegar por las diferentes pantallas principales de la aplicación usando los botones de navegación.

Los botones de navegación se describen en la tabla 6.


Elementos	Función
	Ir a la ventana general, el nivel 0
	Retroceder
	Avanzar
	Ir al nivel superior.

**Tabla 6:** Elementos de navegación de la ventana superior

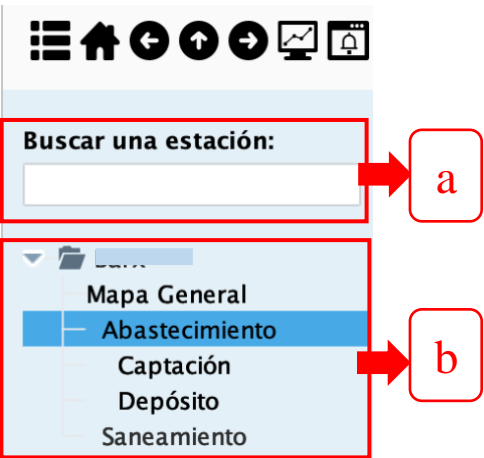
- 3- Acceder a las ventanas de alarmas, históricos y informes, mediante los iconos que se describen en la tabla 7.

Elementos	Función
	Abrir ventana de históricos
	Abrir a la ventana de alarmas
	Generar y abrir el reporte

**Tabla 7:** Elementos adicionales de la ventana superior

- 4- Acceder a la ventana de navegación. El ícono  permite mostrar u ocultar dicha ventana.

La ventana de navegación se expone en la ilustración 41.



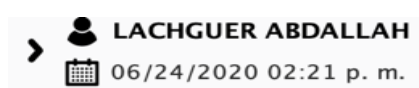
**Ilustración 41:** Ventana de navegación

Como se puede apreciar en la ilustración 39, en la ventana de navegación se han desarrollado dos elementos:

- a- **Buscador de estaciones:** Permite al usuario buscar y abrir la ventana de una estación.
- b- **Árbol de navegación:** Permite al usuario obtener una vista general de la estructura que tienen los mapas principales, así como abrir dichas ventanas. El buscado de estaciones y el árbol de navegación se han desarrollado utilizando el módulo de scripting de *Ignition*.

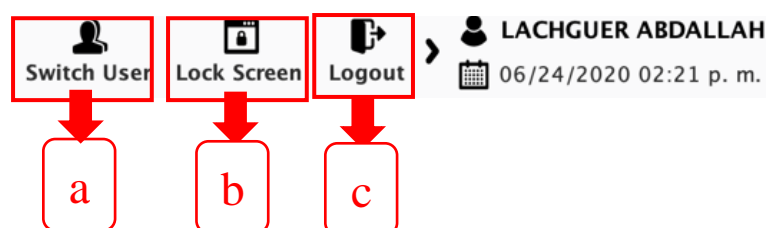
Por seguridad se filtran los datos que se muestran tanto en el árbol de navegación como en el buscador de estaciones según el ámbito o el rol del usuario conectado a la aplicación.

- 5- Mostrar el nombre completo del usuario conectado a la aplicación, así como la información del día y la hora de la conexión. Ver ilustración 42.



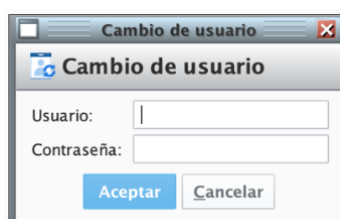
*Ilustración 42: Información del usuario conectado*

- 6- Controles de la aplicación. mediante el ícono (<) el usuario puede mostrar o ocultar un menú de controles de la aplicación. dicho menú se muestra en la figura 43.



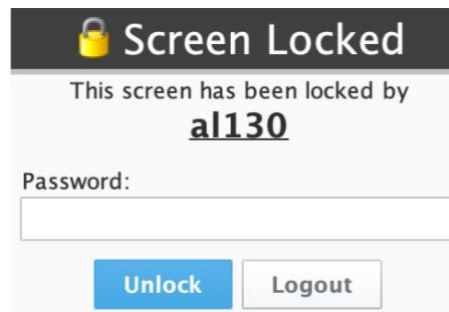
*Ilustración 43: Controles de la aplicación*

- a- **Switch User:** Tiene como función cambiar a otro cliente. Se mostrará una ventana emergente, en la que se deben introducir datos de acceso del nuevo cliente. Ver la ilustración 44.



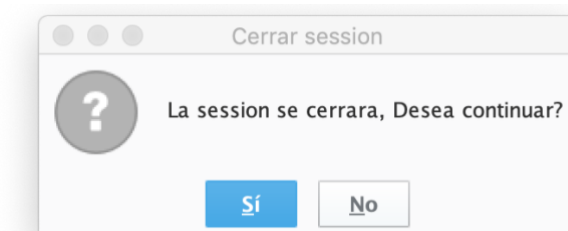
*Ilustración 44: Ventana de cambio de usuario*

- b- Lock Screen:** Permite bloquear la pantalla, cuando se hace clic sobre él, se pierde el control de la pantalla y se mostrará la ventana emergente que se señala en la ilustración 45. En la ventana emergente el usuario debe volver a introducir la contraseña del cliente conectado.



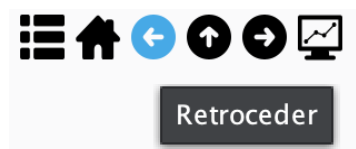
**Ilustración 45:** Ventana de bloqueo de la aplicación

- c- Logout:** Este icono tiene como función cerrar la aplicación. Al ser pulsado se cierra la aplicación. antes de cerrarse aparece el mensaje de confirmación que se presenta en la ilustración 46.



**Ilustración 46:** Mensaje de confirmación de cierre de sesión

Por último, indicar que cuando el cursor del ratón se encuentra sobre cada uno de los botones de la barra superior, se muestra al usuario información de la operación que se a va a realizar. Ver ilustración 47.



**Ilustración 47:** Información sobre los botones de la barra superior




## 2- Ventana inferior

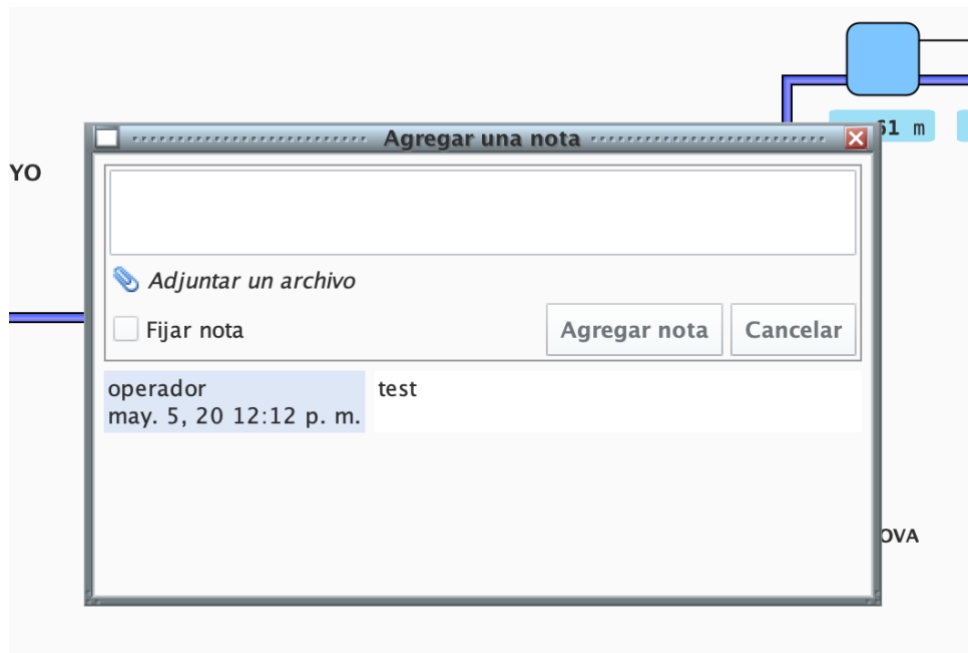
La ventana inferior es una ventana de tipo “*Docked*” con posición al sur de la pantalla. En la ventana inferior se visualiza el estado actual de alarmas según su criticidad. El usuario puede acceder a la ventana de alarmas para ver mas detalles de las alarmas mostradas pulsando sobre cualquiera de los números que aparecen junto con cada tipo de alarma. Las alarmas que se muestran se filtran según el rol o el ámbito del cliente que se conecta a la aplicación.

La ventana inferior tiene el aspecto que se muestra en la ilustración 48.



**Ilustración 48:** Ventana inferior de la aplicación desarrollada.

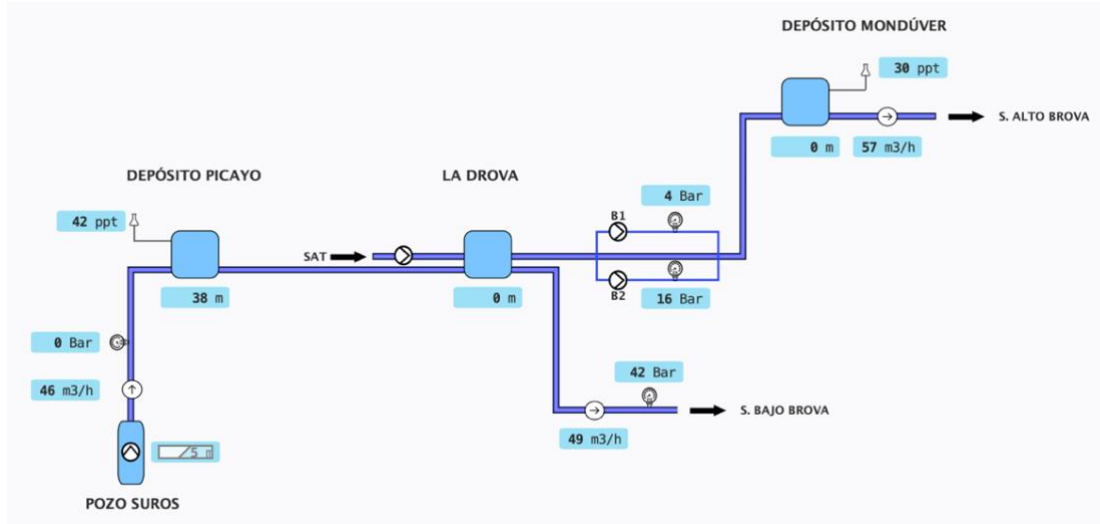
Además de las alarmas, la ventana inferior contiene un panel de comentarios, que permite intercambiar mensajes entre los usuarios conectados a la aplicación. Para acceder al panel de comentarios, se hace clic sobre el icono  situado a la derecha de la ventana. El panel de comentarios se presenta en la ilustración 49.



**Ilustración 49: Panel de comentarios**

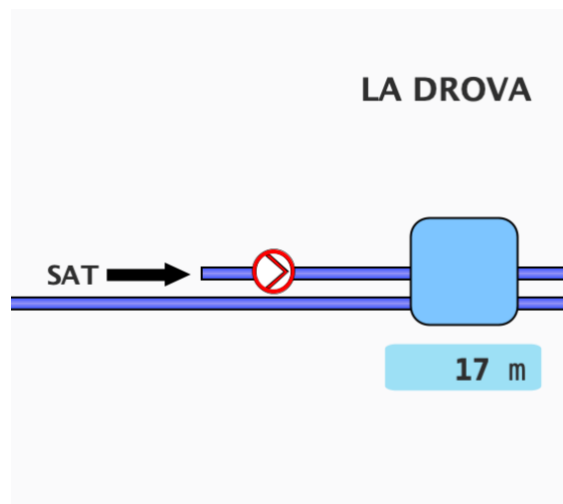
### 3- Ventana principal

La ventana principal es una ventana fija que tiene como función mostrar los sinópticos o los mapas generales. En la ilustración 50 se representa el sinóptico diseñado.



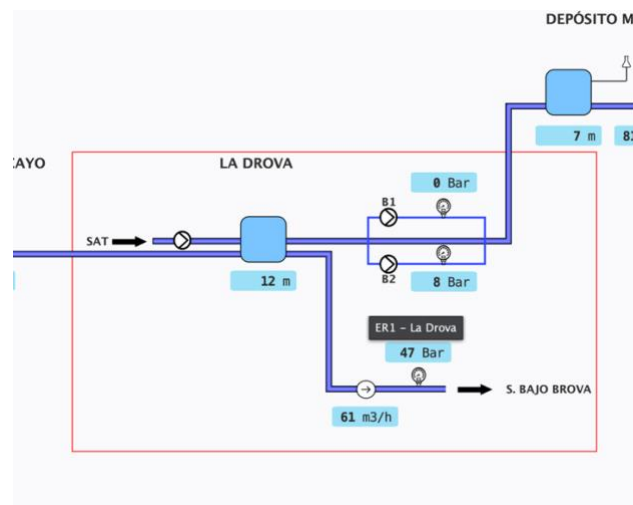
*Ilustración 50: Ventana principal de la aplicación desarrollada*

Dicho mapa muestra al usuario diferentes datos sobre el estado del sistema en general, como se puede observar en el mapa el usuario puede visualizar el valor de las diferentes variables de los elementos que pertenecen al mapa, y también se informa al usuario de las anomalías que presenta el sistema, por ejemplo, el ícono de bomba cambia de color cuando la ocurre algún tipo de fallo. Ver ilustración 51.



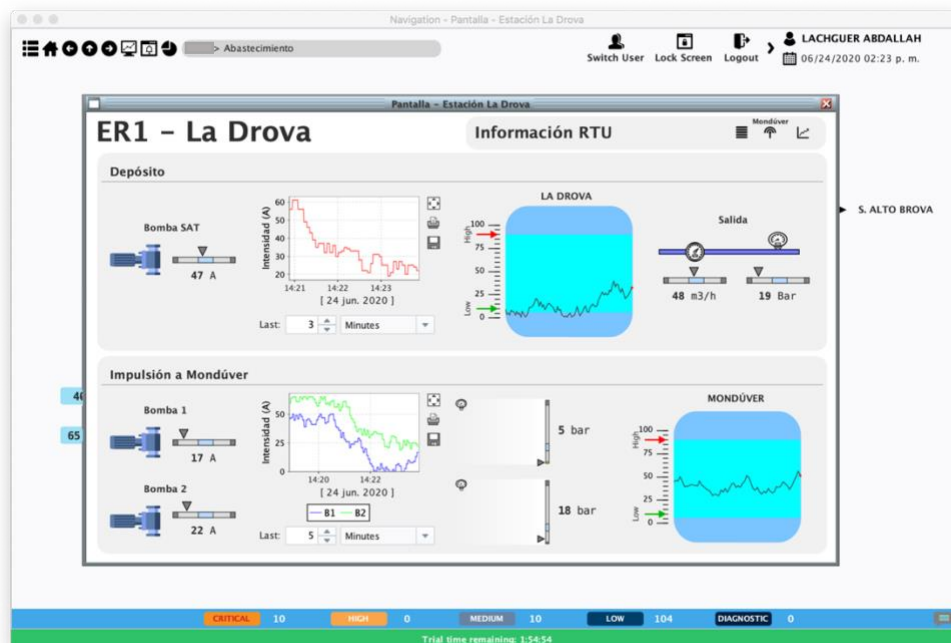
*Ilustración 51: Fallo de una bomba*

Además, desde el sinóptico, el usuario puede acceder a las ventanas de las diferentes estaciones. Cuando se posiciona el cursor del ratón en la zona de estación, aparece un cuadro de color rojo que limita los elementos que contienen la estación. Ver ilustración 52.




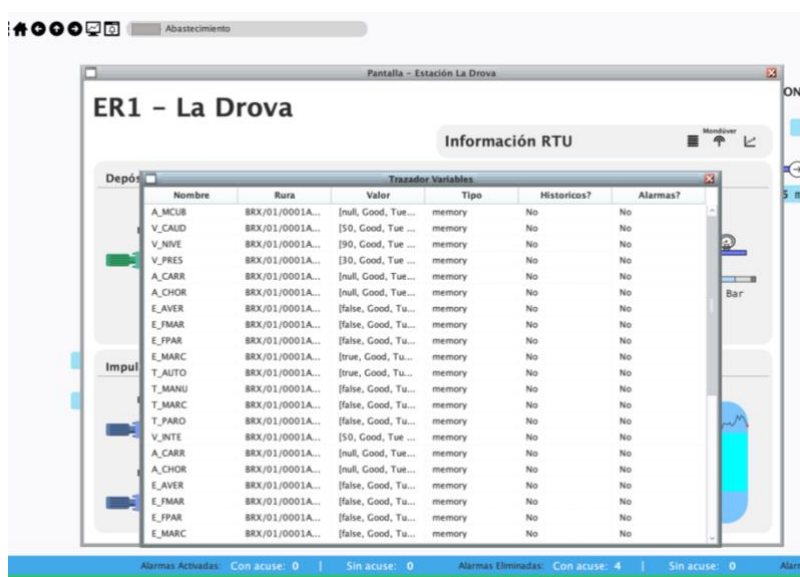
**Ilustración 52:** Estación la Drova

Cuando se hace clic sobre el cuadro rojo aparece la ventana de la estación como un *Popup*. La ventana de estación engloba todas las informaciones de la estación. Se puede ver mas en detalles el estado de todos los elementos. Ver la ilustración 53.



**Ilustración 53:** Ventana de estación

Desde la ventana de estación, se puede acceder al trazador de variables descrito en el apartado: 2.3.4.1, haciendo clic sobre el ícono  situado en la parte superior. El trazador de variables presenta los datos como una tabla. Ver ilustración 54.

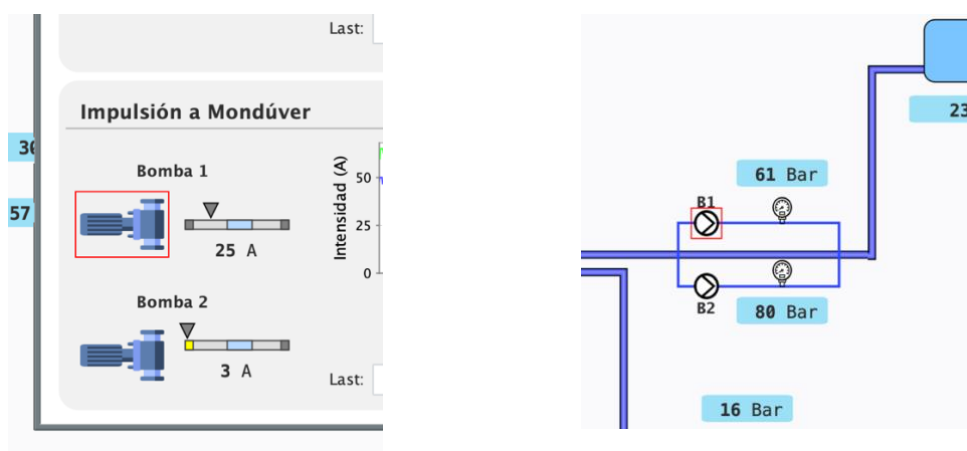


Nombre	Ruta	Valor	Tipo	Historicos?	Alarmas?
A_MCUR	BRX/01/0001A...	[null, Good, Tue...	memory	No	No
V_CAUD	BRX/01/0001A...	[50, Good, Tue ...	memory	No	No
V_NIVE	BRX/01/0001A...	[90, Good, Tue ...	memory	No	No
V_PRES	BRX/01/0001A...	[30, Good, Tue ...	memory	No	No
A_CARR	BRX/01/0001A...	[null, Good, Tue...	memory	No	No
A_CHOR	BRX/01/0001A...	[null, Good, Tue...	memory	No	No
E_AVER	BRX/01/0001A...	[false, Good, Tu...	memory	No	No
E_FMAR	BRX/01/0001A...	[false, Good, Tu...	memory	No	No
E_FPAR	BRX/01/0001A...	[false, Good, Tu...	memory	No	No
E_MARC	BRX/01/0001A...	[true, Good, Tu...	memory	No	No
T_AUTO	BRX/01/0001A...	[true, Good, Tu...	memory	No	No
T_MANU	BRX/01/0001A...	[false, Good, Tu...	memory	No	No
T_MARC	BRX/01/0001A...	[false, Good, Tu...	memory	No	No
T_PARO	BRX/01/0001A...	[false, Good, Tu...	memory	No	No
V_NTE	BRX/01/0001A...	[50, Good, Tue ...	memory	No	No
A_CARR	BRX/01/0001A...	[null, Good, Tue...	memory	No	No
A_CHOR	BRX/01/0001A...	[null, Good, Tue...	memory	No	No
E_AVER	BRX/01/0001A...	[false, Good, Tu...	memory	No	No
E_FMAR	BRX/01/0001A...	[false, Good, Tu...	memory	No	No
E_FPAR	BRX/01/0001A...	[false, Good, Tu...	memory	No	No
E_MARC	BRX/01/0001A...	[false, Good, Tu...	memory	No	No

**Ilustración 54:** Trazador de variables

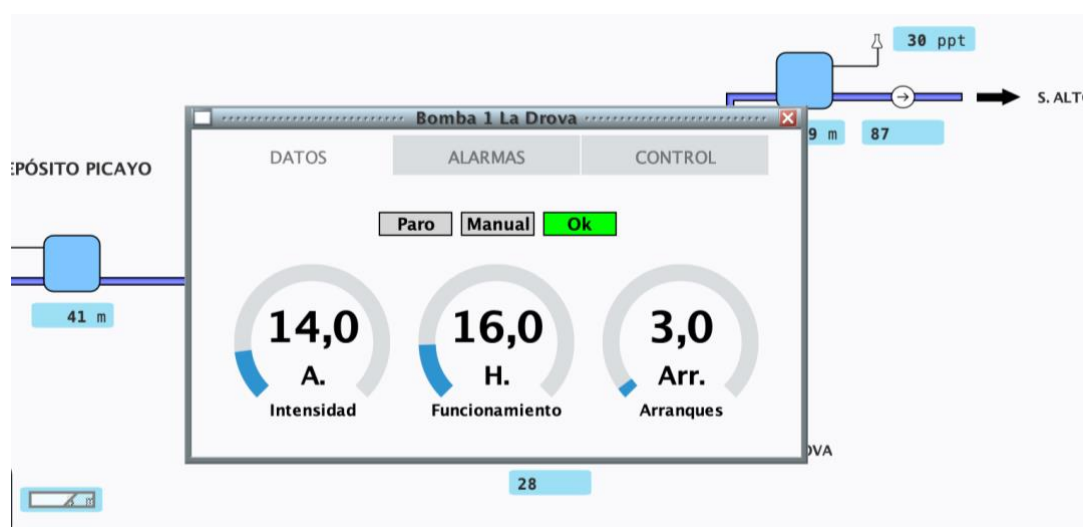
El trazador de variables se ha programado utilizando Scripting de *Ignition*. Para consultar el código utilizado para la programación, referirse al Anexo E: Código trazador de variables

Para acceder a los *faceplates* de los elementos, es posible hacerlo desde el sinóptico o desde la ventana de la estación. Por ejemplo, para acceder al *faceplate* de una bomba desde el sinóptico o desde la ventana de estación se hace clic sobre el ícono de la bomba. Ver la ilustración 55.



**Ilustración 55:** Acceso al *faceplate* de bomba

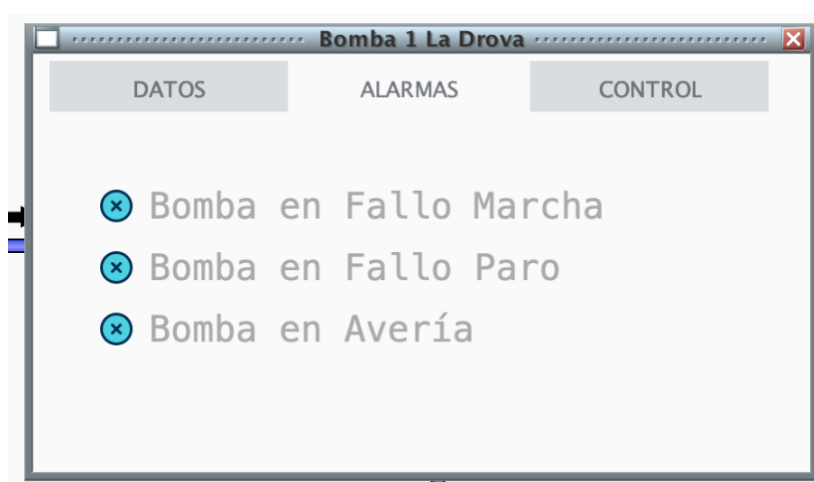
En la ilustración 56 se muestra el *faceplate* de una bomba.



*Ilustración 56: Faceplate de una Bomba*

El *faceplate* de un elemento se compone de tres pestañas, como se puede observar en el ejemplo de la ilustración anterior.

- 1- **Datos:** La pestaña datos sirve para obtener toda la información relacionada con el elemento. En el ejemplo de la bomba, se observa que el usuario puede observar el estado en el que se encuentra la bomba, el modo de funcionamiento. Así como el valor de las distintas variables que contiene la bomba: intensidad, horas de funcionamiento y número de arranques.
- 2- **Alarmas:** En la pestaña alarmas se puede observar si el elemento presenta algún tipo de alarmas. Un ejemplo de la pestaña alarmas se observa en la ilustración 57.



*Ilustración 57: Pestaña alarmas del faceplate de Bomba*

Cuando ocurre una alarma, el título de la pestaña parpadea para indicar al usuario la presencia de una alarma, tal y como se indica en la ilustración 58.



*Ilustración 58: Indicación de alarma en un faceplate de Bomba*

- 3- **Control:** la pestaña control solo existe en los *faceplates* de los actuadores (bombas, válvulas, etc.), sirve para enviar señales de control al elemento. En la ilustración 59 se observa la pestaña control del *faceplate* de una bomba.



*Ilustración 59: Pestaña Control del faceplate Bomba*

Como se puede observar, desde la pestaña control, se puede cambiar el estado de una bomba del modo manual a automático, o viceversa, también permite arrancar o parar la bomba. Todas las acciones de la pestaña control se deben confirmar. Ver la ilustración 60.



*Ilustración 60: Mensaje de confirmación*

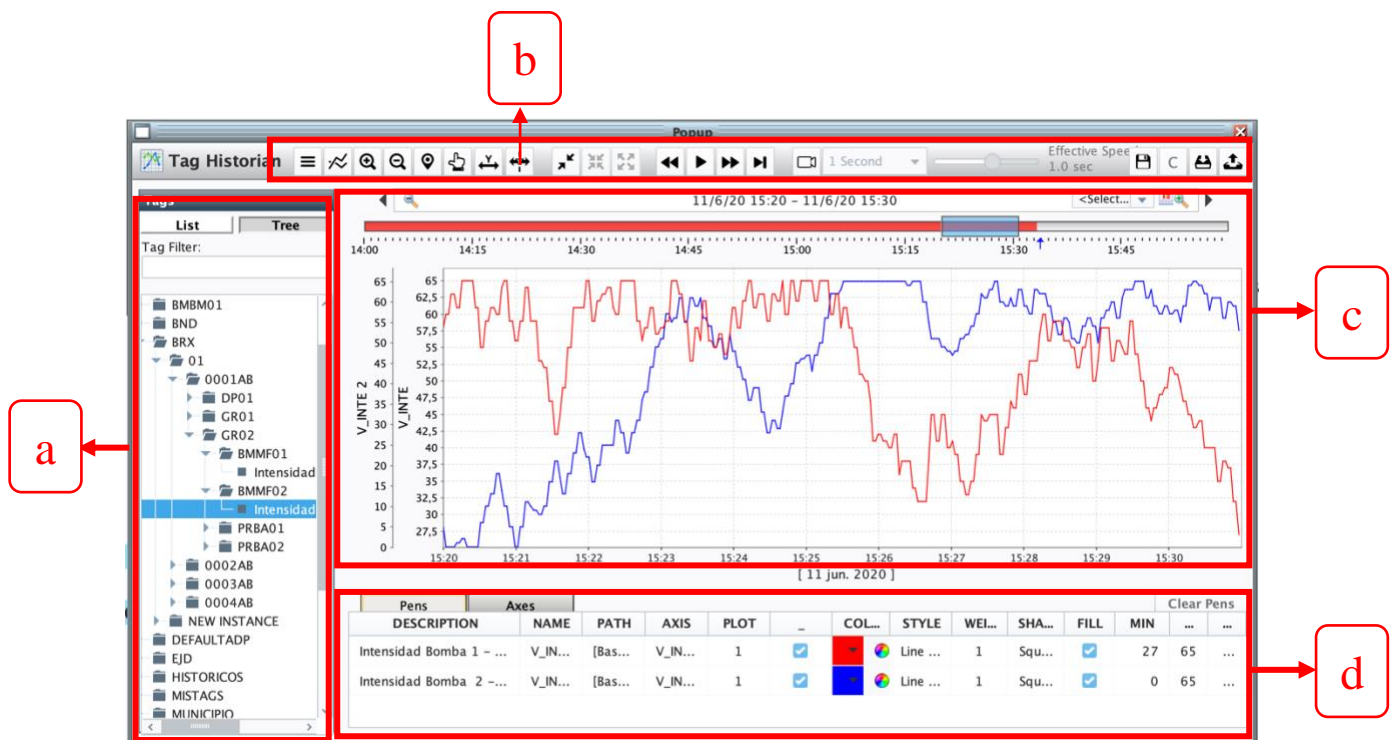
### 3.8.7 Gestión de históricos

Para acceder a la ventana de históricos, en la barra superior, se hace clic en el icono de históricos que se muestra en la ilustración 61.



*Ilustración 61: Acceso a la ventana de históricos*

En la ilustración 62 se ilustra la ventana de históricos. Y se compone de los siguientes elementos:



*Ilustración 62: Ventana de históricos de la aplicación desarrollada*

- a- **Árbol de navegación de tags:** mediante este árbol de navegación, se puede acceder a las etiquetas que se están historizando. Para añadir la etiqueta a la gráfica de visualización, se hace doble clic sobre la misma.
- b- **Barra de opciones:** en esta parte se dispone de varios iconos útiles para interactuar con la gráfica de visualización, entre las opciones se encuentran las siguientes:
  - Separar las gráficas.
  - Aumentar o reducir la vista.

- Cambiar entre visualización en tiempo real y visualización de datos almacenados.
  - Exportar los datos a Excel, etc.
  - Modificar resoluciones de los ejes (tiempos y magnitudes de variable)
  - Realizar zoom en zonas de la ventana.
- c- **Gráficos de visualización:** en esta gráfica se visualizan los históricos de las etiquetas añadidas. En caso de que no esta seleccionada la opción de ver históricos en tiempo real, se dispondrá de un calendario para seleccionar el rango de fechas de los históricos que se mostrarán.
- d- **Tabla de etiquetas añadidas:** en esta tabla se mostrarán informaciones sobre las etiquetas añadidas, como la descripción, el nombre, el *path*, etc. Además, se podrá modificar como se mostrarán en la grafica de visualización, entre las opciones que se pueden modificar, se encuentra el color de la línea, el estilo, el grosor de la línea, etc.

Además de lo descrito anteriormente, la ventana de históricos filtra las etiquetas que se mostrarán en el árbol de navegación según el rol o el ámbito del usuario que se conecta a la aplicación.

### **Integración de datos históricos procedentes de *dataloggers* en la base de datos:**

Los *dataloggers* o registradores de datos son unos dispositivos electrónicos que registran datos en el tiempo o en relación a la ubicación por medio de instrumentos o sensores propios o conectados externamente. Los *dataloggers* envían archivos *CSV* (*Comma-Separated Values*) con los datos registrados. En la aplicación de *Wonderware* se leen dichos archivos de forma periódica y se insertan en los datos históricos. En *Ignition* la solución adoptada fue programar utilizando el *Scripting* de *Ignition* para abrir los archivos *CSV*, leerlos y insertarlos en la base de datos.

Para consultar el código utilizado para la integración de los datos procedentes de *dataloggers* en los históricos, referirse al Anexo F: Código Integración de *dataloggers*.

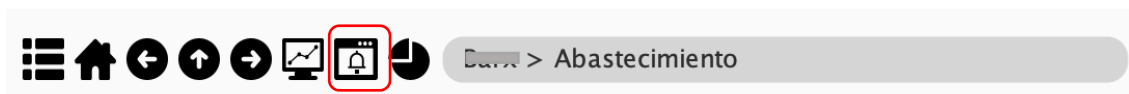


### 3.8.8 Gestión de Alarmas

En las etiquetas que llevan la propiedad de alarmas, Se han configurado cuatro alarmas:

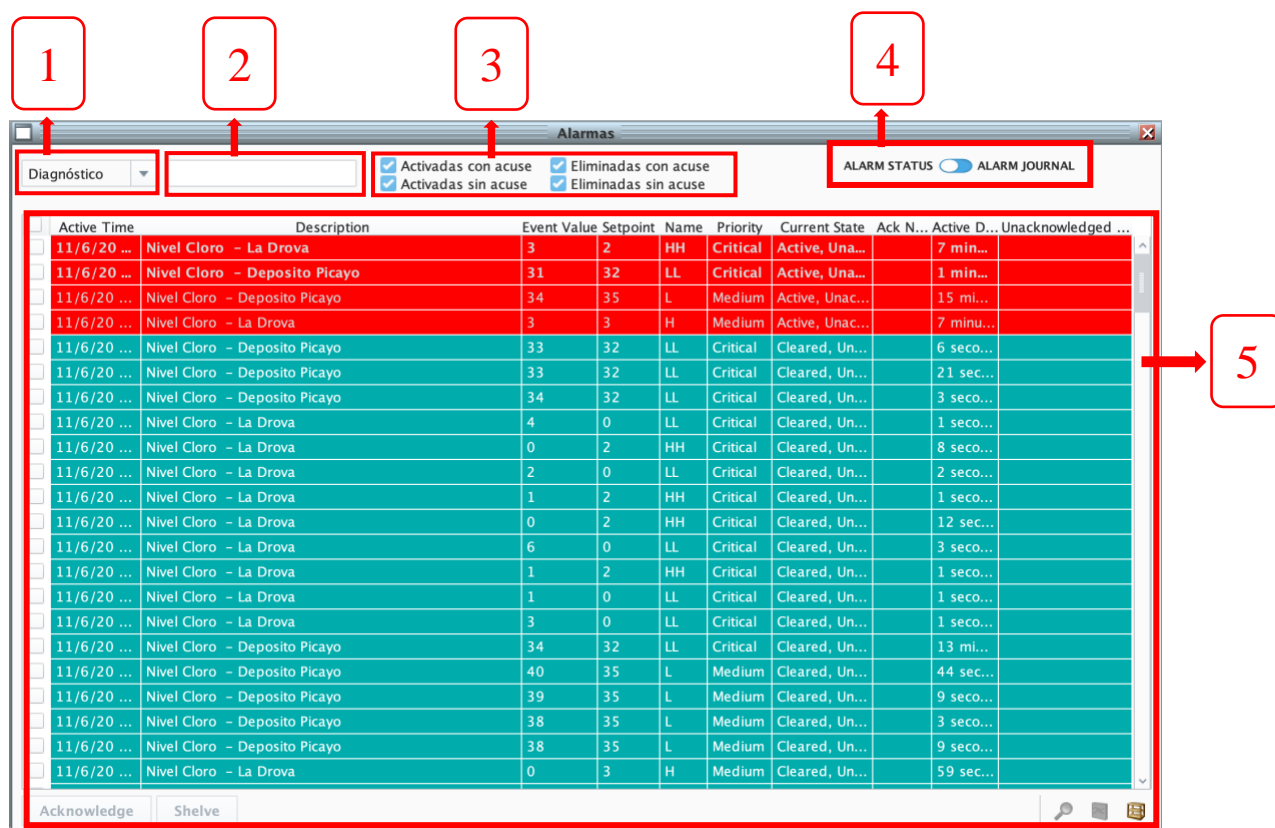
- HH: Nivel muy alto
- H: Nivel alto
- L: Nivel bajo
- LL: Nivel muy bajo

Para acceder a la ventana de gestión de alarmas se pulsa sobre el icono de alarmas que se representa en la ilustración 63.



*Ilustración 63: Acceso a la ventana de alarmas*

En la ilustración 64 se muestra la ventana de gestión de alarmas diseñada.



*Ilustración 64: Ventana de alarmas de la aplicación desarrollada*

- 1- Una lista para filtrar las alarmas por la criticidad mínima, se puede elegir entre:
  - a. Diagnostico
  - b. Bajo
  - c. Medio
  - d. Alto
  - e. Crítico
- 2- Un filtro tipo “entrada de texto” para filtrar las alarmas que se mostrarán.
- 3- Otro filtro para mostrar las alarmas según su estado. Los estados pueden ser:
  - a. Alarmas activadas y reconocidas
  - b. Alarmas activadas y no reconocidas
  - c. Alarmas eliminadas y reconocidas
  - d. Alarmas eliminadas y no reconocidas.
- 4- Un selector para cambiar entre la tabla de alarmas en tiempo real y la tabla de histórico de alarmas.
- 5- Tabla de alarmas.
  - a. Tabla de alarmas en estado real.

En la ilustración 65 se indica la tabla de alarmas en estado real.

Active Time	Description	Event Value	Setpoint	Name	Priority	Current State	Ack N...	Active D...	Unacknowledged ...
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	0	0	LL	Critical	Active, Una...		9 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	0	1	L	Medium	Active, Unac...		9 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - Deposito Picayo	34	32	LL	Critical	Cleared, Un...		3 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - Deposito Picayo	34	32	LL	Critical	Cleared, Un...		13 mi...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - Deposito Picayo	35	32	LL	Critical	Cleared, Un...		1 minu...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - Deposito Picayo	38	32	LL	Critical	Cleared, Un...		3 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	0	2	HH	Critical	Cleared, Un...		4 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	5	0	LL	Critical	Cleared, Un...		4 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - Deposito Picayo	35	32	LL	Critical	Cleared, Un...		6 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	0	2	HH	Critical	Cleared, Un...		5 minu...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	6	0	LL	Critical	Cleared, Un...		1 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	0	2	HH	Critical	Cleared, Un...		2 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	2	0	LL	Critical	Cleared, Un...		5 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	0	2	HH	Critical	Cleared, Un...		3 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	3	0	LL	Critical	Cleared, Un...		1 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	0	2	HH	Critical	Cleared, Un...		1 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - Deposito Picayo	40	35	L	Medium	Cleared, Un...		15 mi...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - Deposito Picayo	38	35	L	Medium	Cleared, Un...		6 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - La Drova	5	1	L	Medium	Cleared, Un...		4 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - Deposito Picayo	37	35	L	Medium	Cleared, Un...		3 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - Deposito Picayo	37	35	L	Medium	Cleared, Un...		3 seco...	
<input type="checkbox"/> 11/6/20 ...	Nivel Cloro - Deposito Picayo	36	35	L	Medium	Cleared, Un...		12 sec...	

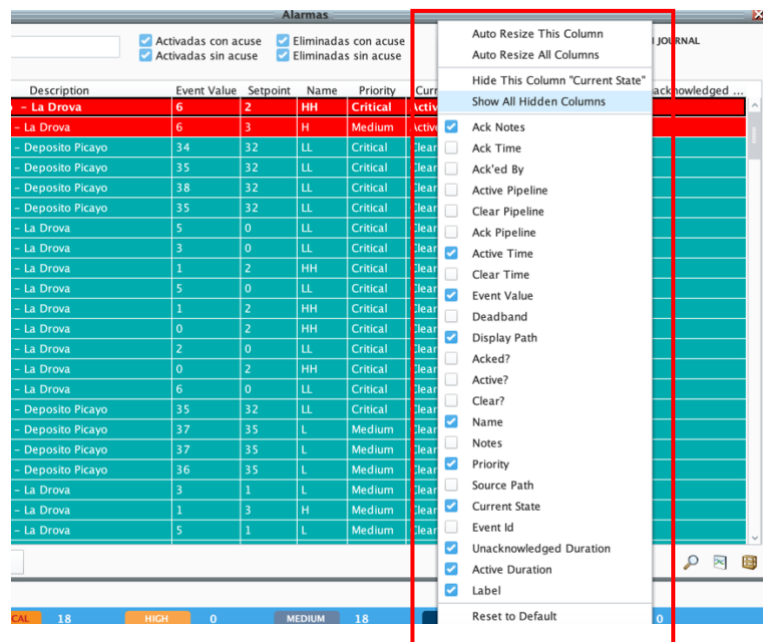
**Ilustración 65:** Tabla de alarmas en estado real

Esta tabla muestra el estado actual de todas las alarmas del sistema. Cada alarma está codificada por colores para que puede identificar rápidamente su estado.

La tabla esta configurada para mostrar las siguientes informaciones:

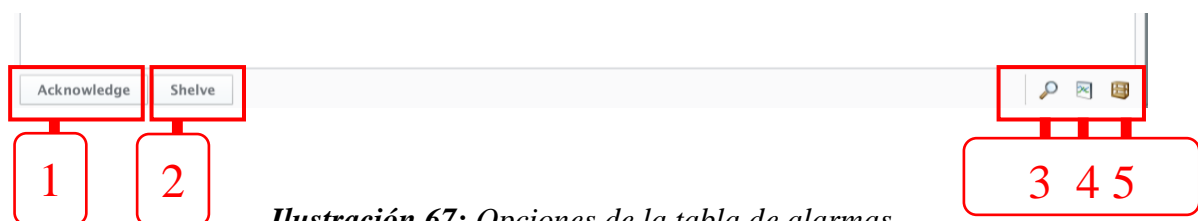
- Tiempo de activación de alarma.
- Descripción de la etiqueta en la que se ha producido la alarma.
- El valor de la etiqueta que ha producido la alarma.
- El *Setpoint*.
- El nombre de la alarma
  - *High High*
  - *High*
  - *Low*
  - *Low Low*
- Prioridad de la alarma
  - *Critical*
  - *High*
  - *Medium*
  - *Low*
  - *Diagnostic*
- Estado actual de la alarma
  - Activada y reconocida
  - Activada y no reconocida
  - Eliminada y reconocida
  - Eliminada y no reconocida
- Notas
- Tiempo que llevan activadas las alarmas.

La tabla es personalizable, ya que el usuario puede modificar las columnas que se muestran en la tabla, haciendo clic con el botón derecho en la tabla. Como se señala en la ilustración 66.



**Ilustración 66:** Personalizar la tabla de alarmas

En la parte inferior de la tabla, se dispone de las siguientes opciones que se indican en la ilustración 67.

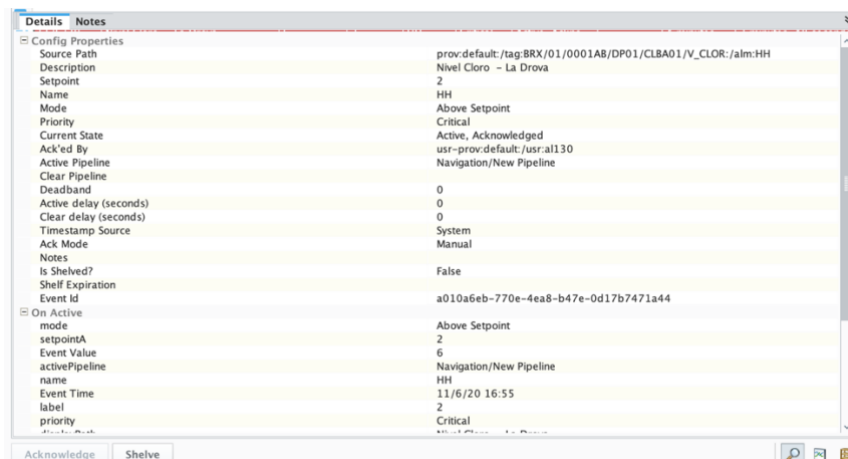


**Ilustración 67:** Opciones de la tabla de alarmas

Dichas opciones permiten lo siguiente:

- 1- Reconocer la alarma seleccionada.
- 2- Posponer la alarma seleccionada: mediante esta opción se puede posponer una alarma un determinado tiempo, dicha alarma desaparecerá de alarma durante el tiempo seleccionado.

- 3- Obtener todos los detalles de la alarma seleccionada. Ver la ilustración 68.



**Ilustración 68:** Detalles de alarma seleccionada

- 4- Ver histórico de la etiqueta en la que se produce la alarma seleccionada. El histórico se muestra en la ilustración 69.



**Ilustración 69:** Histórico de la alarma seleccionada

- 5- Acceder a las alarmas pospuestas: sirve para ver las alarmas pospuestas, se presentan la tabla que se muestra en la ilustración 70 muestra el usuario que pospuso la alarma y el tiempo restante, además se puede dejar de posponer la alarma.

Path	Shelved By	Expires
prov:default:/tag:BRX/01/0001AB/DP01/CLBA01/V_CLOR:/alm:HH	a1130	4 minutes, 43 seconds

Unshelve

**Ilustración 70:** Ventana de "Shelved Alarms"

- b. Tabla de histórico de alarmas: Permite explorar el historial de alarmas para el rango de tiempo seleccionado. La tabla de histórico se muestra en la ilustración 71.

Event Time	Event Id	Description	Event State	Priority	System	Ack'd By	Event Value	Current State	Setpoint
11/6/20 17:31	1508df73-...	Nivel Caudal - M...	Clear	Low	False		85	Cleared, U...	80.0
11/6/20 17:31	1508df73-...	Nivel Caudal - M...	Active	Low	False		80	Active, Una...	80.0
11/6/20 17:31	60612e76-...	Nivel Caudal - M...	Ack	Low	False	Live Event L...		Cleared, Ac...	H
11/6/20 17:31	5af9354b-...	Nivel Deposito - ...	Clear	Low	False		87	Cleared, U...	80.0
11/6/20 17:30	5af9354b-...	Nivel Deposito - ...	Active	Low	False		80	Active, Una...	80.0
11/6/20 17:30	643ce323-...	Nivel Deposito - ...	Ack	Low	False	Live Event L...		Cleared, Ac...	H
11/6/20 17:30	1cdbae68-...	Nivel Caudal - La...	Clear	Low	False		4	Cleared, U...	2.0
11/6/20 17:30	1cdbae68-...	Nivel Caudal - La...	Active	Low	False		2	Active, Una...	2.0
11/6/20 17:30	4ad6f92a-...	Nivel Caudal - La...	Ack	Low	False	Live Event L...		Cleared, Ac...	L
11/6/20 17:30	ede6c1d0-...	Nivel Caudal - La...	Clear	Low	False		2	Cleared, U...	1.0
11/6/20 17:30	ede6c1d0-...	Nivel Caudal - La...	Active	Low	False		1	Active, Una...	1.0
11/6/20 17:30	46bf3144-...	Nivel Caudal - La...	Ack	Low	False	Live Event L...		Cleared, Ac...	LL
11/6/20 17:30	1eb19769-...	Nivel Caudal - La...	Clear	Low	False		7	Cleared, U...	1.0
11/6/20 17:30	1eb19769-...	Nivel Caudal - La...	Active	Low	False		1	Active, Una...	1.0
11/6/20 17:30	fa1fc336-f...	Nivel Caudal - La...	Ack	Low	False	Live Event L...		Cleared, Ac...	LL
11/6/20 17:30	fd45ce9a-...	Nivel Caudal - La...	Clear	Low	False		4	Cleared, U...	1.0
11/6/20 17:30	fd45ce9a-...	Nivel Caudal - La...	Active	Low	False		1	Active, Una...	1.0
11/6/20 17:30	a37d2143-...	Nivel Caudal - La...	Ack	Low	False	Live Event L...		Cleared, Ac...	LL
11/6/20 17:29	22f7dce4-...	Presion - La Drova	Clear	Low	False		7	Cleared, U...	0.0
11/6/20 17:29	0f4d27bd-...	Presion - La Drova	Clear	Low	False		7	Cleared, U...	0.0
11/6/20 17:29	ed02b694-...	Presion - La Drova	Clear	Low	False		7	Cleared, U...	0.0
11/6/20 17:29	c097d633-...	Presion - La Drova	Clear	Low	False		7	Cleared, U...	0.0

7.509 events

**Ilustración 71:** Tabla de históricos de alarmas

- 1- Permite exportar los datos a una tabla Excel.
- 2- Permite seleccionar el rango de fechas.
- 3- Seguir una alarma específica.
- 4- Permite ver toda la información asociada con la alarma seleccionada.
- 5- Permite filtrar las alarmas que se muestran. Ver ilustración 72.

Filter

Search

Priority: Diagnostic

System Events: ☒ true

Active Events: ☒ true

Cleared Events: ☒ true

Acked Events: ☒ true

**Ilustración 72:** Ventana de filtros de la tabla de alarma

### 3.8.9 Notificación de alarmas

Los tipos de notificaciones posibles en *Ignition* son los siguientes (Inductive Automation, 2020):

- Por correo electrónico:
  - Envía un correo electrónico a un grupo de personas cuando una alarma se activa o se borra.
  - Los operadores también pueden reconocer la alarma haciendo clic en un enlace que se envía en el cuerpo del correo electrónico.
- Por SMS:
  - Permite enviar notificaciones de alarma por SMS a través de un modem celular configurado con una tarjeta SIM.
  - Los operadores pueden responder con un código especial para reconocer la alarma.
  - El módulo SMS depende del módulo de notificación de alarma. Deber tener ambos módulos instalados.
- Por Voz
  - Realice llamadas de voz a través de cualquier sistema telefónico compatible con SIP (*Voip*)
  - Admite varios idiomas al mismo tiempo, según las preferencias del usuario.
  - Permite a los usuarios reconocer eventos.
  - Soporta requerir un número de identificación personal por temas de seguridad.
  - Se vincula al registro de auditoría para auditar eventos de llamadas, entrega exitosa de mensajes y acuses de recibo de los usuarios.
  - Soporta consolidación de mensaje.

Como ya se ha mencionado en el apartado 2.3.4.1, para la notificación de alarmas, Suez utiliza Módulo llamado *Nuntius*. *Nuntius* es un cliente OPC UA que obtiene las propiedades de alarmas para luego notificar a los usuarios.

En la versión actual de *Ignition* no se puede utilizar *Nuntius*, ya que el servidor OPC UA de *Ignition* solo proporciona valores de *tags* en tiempo real, y no permite consultar sus propiedades tal como alarmas. Como alternativa a *Nuntius*, en el presente proyecto se utiliza el módulo de *Alarm Notification* de *Ignition*.

Para configurar la notificación de alarmas por correo electrónico, primero se debe configurar el servidor de salida de correo. Para ello se accede al Gateway y luego se dirige a la siguiente ruta: **Configuration > Alarming > Notification > Create new Alarm Notification Profile**, Luego se selecciona **Email Notification**, y por último configurar el servidor SMTP. Ver ilustraciones 73 y 74.

☒ **Email Notification**  
Send alarm notifications via email.

☐ **Remote Gateway Notification**  
Enables alarm notification using notification profiles of a remote gateway. Will also expose the pipelines of the remote gateway directly to tags.

☐ **Simple One-way Email Notification**  
Send alarm notifications via email to users in a user source. Listens for active alarms within a given priority range.

☐ **SMS Notification**  
Send alarm notifications via SMS using your Airlink device.

[Next >](#)

**Ilustración 73: Notificación por correo electrónico**

**Email Settings**

Use SMTP Profile? ☐ If selected, this notification profile will use one of the gateway defined SMTP profiles. Otherwise, it will use the settings defined here. (default: false)

SMTP Profile:  If Use SMTP Profile is selected, alarm notifications will be emailed using this profile.

Hostname:  Hostname of the SMTP server to send email through. Only used when Use SMTP Profile is false.

Port:  Port SMTP service is running on. Only used when Use SMTP Profile is false. (default: 25)

Enable SSL/TLS: ☐ Connect using SSL/TLS. Only used when Use SMTP Profile is false. (default: false)

Username:  Only used when Use SMTP Profile is false.

Change Password? ☐ Check this box to change the existing password.

Password:

Password:  Re-type password for verification.

[Save Changes](#)

**Ilustración 74: Configuración del servidor SMTP**

El siguiente paso es configurar el grupo de usuarios que se va a notificar. Estando en el Gateway, se accede a la siguiente ruta: **Configuration > Alarming > Notification > On call Rosters > Create new On call Roster**. Ver ilustración 75.

**Properties**

Name:

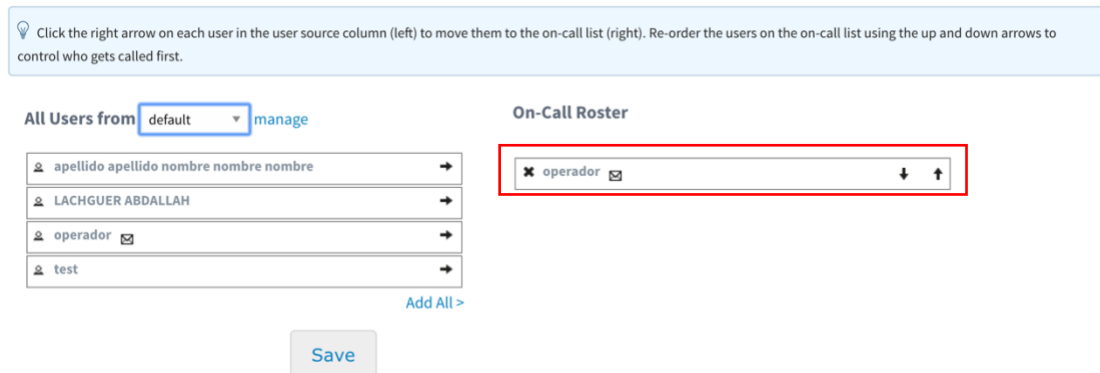
Description:

[Save Changes](#)

**Ilustración 75: Configuración del grupo de usuarios**



Una vez configurado el grupo de usuarios, en opciones de dicho grupo, se añaden los usuarios que se quiere notificar, en este caso se añade solo un usuario, llamado Operador. Ver ilustración 76.



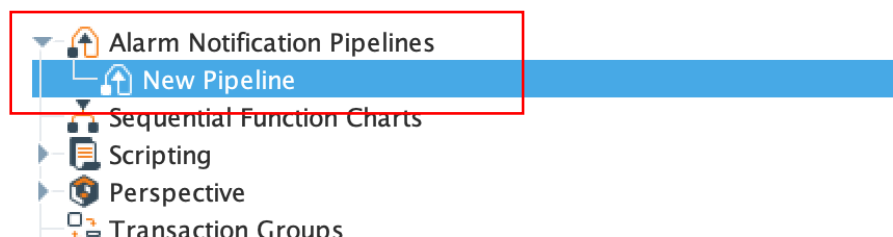
**Ilustración 76:** Añadir usuario al grupo de usuarios

La información de correo electrónico se añade en el perfil de usuario. Como se puede ver en la ilustración 77.



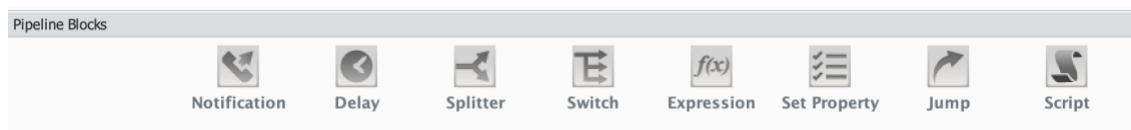
**Ilustración 77:** Información de correo electrónico de un usuario

El siguiente paso es crear una lógica de notificación. En el software “**Designer**”, en **Alarm Notification Pipelines**, se crea un nuevo *Pipeline* (lógica de notificación). Tal y como se muestra en la ilustración 78.



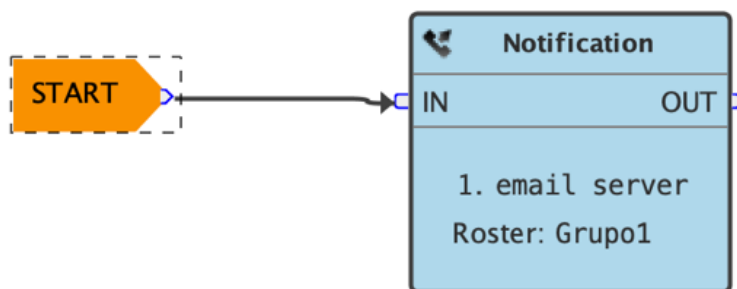
**Ilustración 78:** Crear un nuevo New Pipeline

Para diseñar la lógica de notificación, se dispone de varios bloques de funciones. Ver la ilustración 79.



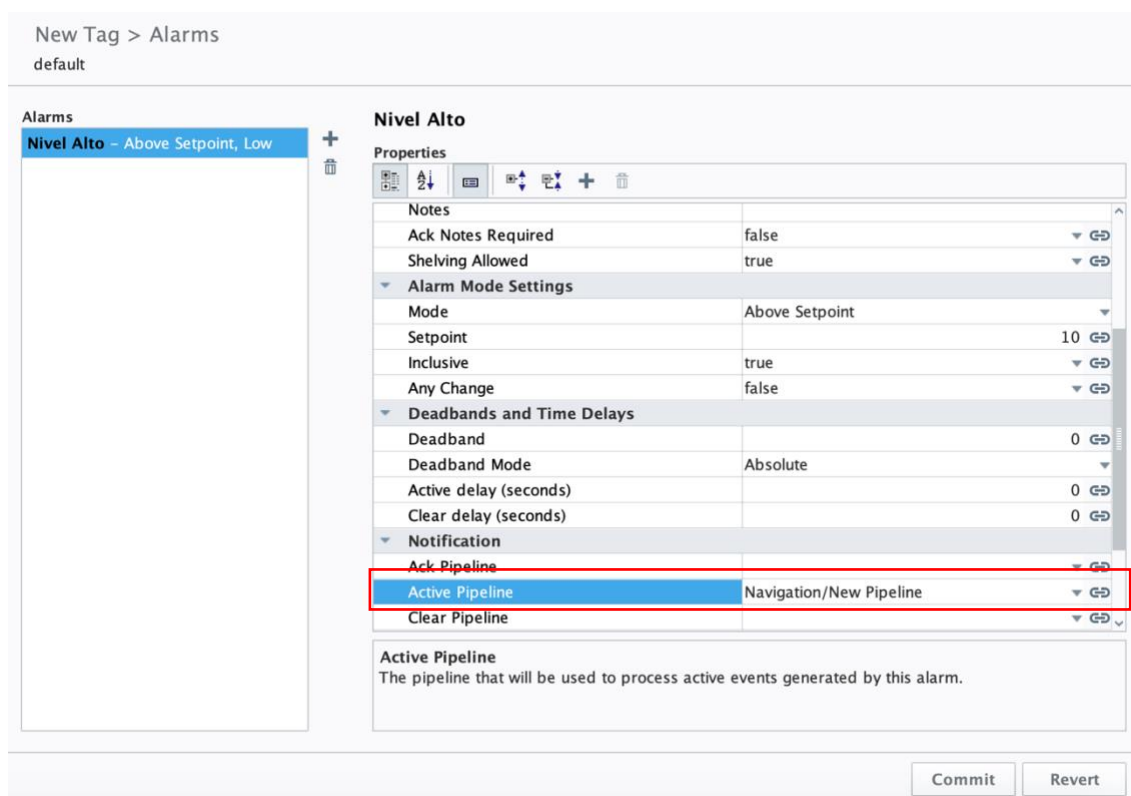
**Ilustración 79:** Bloques de funciones en la lógica de notificación

La lógica de notificación diseñada tiene el aspecto que se muestra en la ilustración 80.



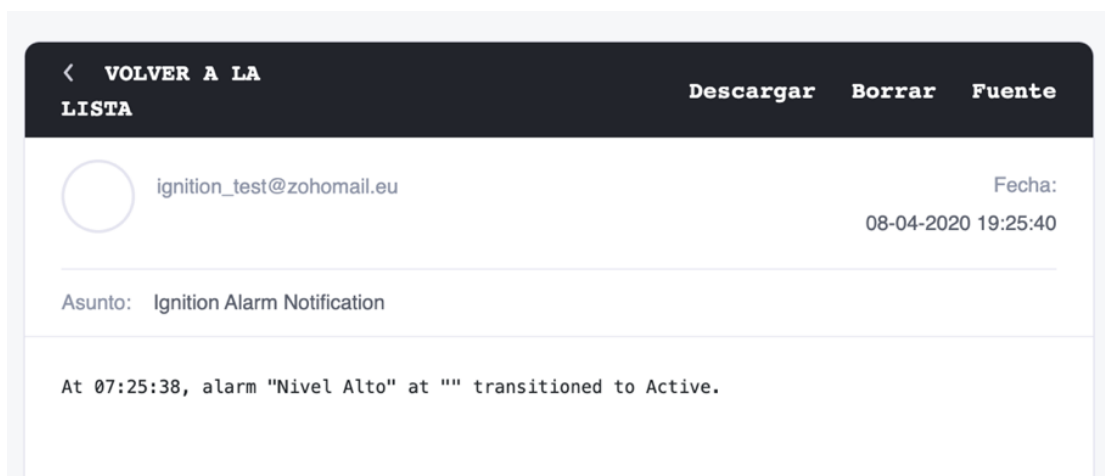
**Ilustración 80:** Lógica de notificación

Por último, se configuran los *Tags* para que se notifiquen con el Pipeline creado. Ver la ilustración 81.



**Ilustración 81:** Configuración de tags con el Pipeline

Los correos de notificación de alarmas se reciben correctamente. Y tiene el formato que se muestra en la ilustración 82.



***Ilustración 82: Recepción de correo de notificación de alarmas***



# 4 Capitulo 4: Presupuesto, Conclusiones y Trabajos futuros

## 4.1 Presupuestos

### 4.1.1 Presupuesto Mínimo Producto Viable

El presente presupuesto hace referencia al tiempo empleado en el desarrollo de todas las actividades planteadas en el presente proyecto. El desarrollo de cada actividad consiste en el análisis de los requisitos, diseño, programación, pruebas y documentación.

	Perfil	€-JORNADA
	DJ Desarrollador junior	256,00 €
	DS Desarrollador/arquitecto senior	360,00 €
	DP Dirección de proyecto	480,00 €

Presupuesto desarrollo TFM	DJ (Días)	DS (Días)	DP (Días)	Total
Análisis plataforma WW	10	2	0,6	3.568,00 €
Planificación técnica del proyecto	2	1,5	0,175	1.136,00 €
Plantillas de objetos gráficos y UDT's	8	1	0,45	2.624,00 €
Automatizar la creación de instancias	8	1	0,45	2.624,00 €
Creación de sinópticos y mapas de proceso	8	1	0,45	2.624,00 €
Trazador de variables	8	1	0,45	2.624,00 €
Estructura de aplicación SCADA y navegación básica	8	1	0,45	2.624,00 €
Gestión de históricos	8	1	0,45	2.624,00 €
Gestión de alarmas	8	1	0,45	2.624,00 €
Integración de Nuntius y Dataloggers	8	1	0,45	2.624,00 €
Árbol de navegación y buscador de estaciones	8	1	0,45	2.624,00 €
Total=>				28.320,00 €

*Tabla 8: Presupuesto elaboración TFM*

#### 4.1.2 Presupuesto desarrollo de una plataforma global en *Ignition*

El presente presupuesto hace referencia a una estimación económica para el desarrollo de una plataforma global con el software *Ignition*.

Uds	CONCEPTO	DJ (Días)	DS (Días)	DP (Días)	Total
141	Plantilla Simple (5 tags)	150			38.400,00 €
33	Plantilla Compleja (10 tags)	80			20.480,00 €
21	Plantilla Avanzada (> 10 tags)	80			20.480,00 €
1	Configurador de instancias	50			12.800,00 €
1	Trazador de variables	20			5.120,00 €
1	Navegación, árbol y buscador	20			5.120,00 €
1	Seguridad y control de usuarios	10			2.560,00 €
1	Controles de Historicos	20			5.120,00 €
1	Controles de Alarmas	10			2.560,00 €
1	Integración N3uron	10			2.560,00 €
1	Integración Nuntius	10			2.560,00 €
1	Integración DL Sofrel	10			2.560,00 €
1	Integración DL Mejoras	10			2.560,00 €
1	Integración DL Microcom	10			2.560,00 €
10	Widgets Avanzados	50			12.800,00 €
1	Infraestructura de servidores y redundancia	10			2.560,00 €
1	Seguimiento técnico		60		21.600,00 €
1	Seguimiento de proyecto, planificación			30	14.400,00 €
1	Licencias Ignition				15.300,00€
	<b>Total</b>				<b>192.100,00 €</b>
	PRESUPUESTO MINIMO PRODUCTO VIABLE				28.320,00 €
	PRESUPUESTO DESARROLLO PLATAFORMA				192.100,00 €
	<b>TOTAL</b>				<b>192.100,00 €</b>
	<b>ESTIMACIÓN DE AHORRO SUEZ POR REALIZACIÓN DEL PROYECTO</b>				<b>16.992,00 €</b>

**Tabla 9:** Presupuesto desarrollo de una plataforma global con *Ignition*

El presupuesto estimado para desarrollar una plataforma global con el software *Ignition* en Suez es 192.100,00 euros, con un ahorro aproximado a 16.992,00 euros debido a la realización de este proyecto.

## 4.2 Conclusiones y Trabajos futuros

En el presente trabajo se presenta una aplicación SCADA programada con la tecnología *Ignition* con el objetivo de estudiar las características de dicha plataforma y compararla con la tecnología *Wonderware*.

La implementación y el diseño de la aplicación SCADA ha permitido aprender a utilizar todas las herramientas de la plataforma *Ignition*.

Todos los módulos analizados de la aplicación *Wonderware* se han podido desarrollar en *Ignition* excepto el módulo de *Nuntius*, la solución adoptada para dicho módulo ha sido utilizar el módulo de notificación de alarmas de *Ignition*.

Durante el desarrollo de actividades del presente proyecto, y vistos los resultados obtenidos, se ha podido comprobar que la tecnología *Ignition* es una tecnología muy competitiva y puede ser una correcta opción a nivel de funcionalidades, potencia y robustez para sustituir a la plataforma *Wonderware* en la empresa Suez. Además, *Ignition* cuenta con la gran ventaja de ser una plataforma basada en Java y no depende del sistema operativo, así como disponer de una licencia ilimitada, y con esto se puede tener un número ilimitado de clientes, pantallas, *tags*, conexión y dispositivos, lo que desde el punto de vista económico significa una disminución de costes para la empresa comparado con *Wonderware* u otro tipo de tecnología.

En cuanto a trabajos futuros queda pendiente la evaluación de la aplicación desarrollada siguiendo estándares y normativas relacionadas a Interfaz Humano-Máquina (HMI), para optimizar la aplicación y detectar los aspectos y puntos que se deben mejorar. Sería interesante también realizar pruebas con *Ignition* a nivel de ciberseguridad y acceso a la información, y por último realizar una aplicación completa para estaciones reales haciendo las conexiones a PLCs e instrumentos de campo, y realizar pruebas sobre la aplicación. Dichas pruebas pueden ser uso masivo de la aplicación mediante la conexión de varios clientes a la vez y la comunicación con un número grande de variables.





## 5 Bibliografía

- Borghello, C. (12 de 06 de 2017). *Ciberseguridad en la Pirámide de Automatización Industrial*. Obtenido de Segu-Info: <https://blog.segu-info.com.ar/2017/06/ciberseguridad-en-la-piramide-de.html>
- Inductive Automation. (2020). <https://docs.inductiveautomation.com/>. Obtenido de Ignition 8 User's Manual : <https://docs.inductiveautomation.com/>
- Kymera systems. (2020). *KYMER MAP PANEL MODULE*. Obtenido de Kymera Systems: <https://www.kymerasystems.com/kymera-map-panel-module/>
- Oracle. (2020). *Base de datos*. Obtenido de Oracle: <https://www.oracle.com/es/database/what-is-database.html>
- Penin, A. R. (2013). *Sistemas SCADA*. México: Alfaomega Grupo Editor, S.A de C.V.



## 6 Anexos

### Anexo A: Código del generador de instancias

```
1. municipio=event.source.parent.getComponent('Municipio').text
2. servicio=event.source.parent.getComponent('Servicio').selectedStringValue
3. estacion=event.source.parent.getComponent('Estacion').text
4. grupo=event.source.parent.getComponent('Grupo').text
5.
6. BombaNum=event.source.parent.getComponent('BombaNum').intValue
7. CloroNum=event.source.parent.getComponent('CloroNum').intValue
8. PresionNum=event.source.parent.getComponent('PresionNum').intValue
9. NivelNum=event.source.parent.getComponent('NivelNum').intValue
10.
11. if event.source.parent.getComponent('Servicio').selectedValue==0:
12.     CS=str(1)
13.     CS=(CS.zfill(2))
14. if event.source.parent.getComponent('Servicio').selectedValue==1:
15.     CS=str(2)
16.     CS=(CS.zfill(2))
17. path=municipio+'/'+municipio+CS+'/'+municipio+CS+estacion+servicio+'/'+municipio+
    CS+estacion+servicio+"_GR"+grupo
18.
19. if event.source.parent.getComponent('bomba').selected:
20.     if event.source.parent.getComponent('TiposBomba').selectedStringValue == "BMM
        F_Bomba_Mando_UDAs_Fallos":
21.         nam="_BMMF0";
22.         for i in range(BombaNum):
23.             system.tag.addTag(parentPath=path,
24.                 name=municipio+CS+estacion+servicio+"_GR"+grupo+nam+str(i+1),
25.                 tagType="UDT_INST",
26.                 attributes={"UDTParentType": "Bomba/BMMF_Bomba_Mando_UDAs_Fallos"},
27.             )
28. if event.source.parent.getComponent('cloro').selected:
29.     if event.source.parent.getComponent('TiposCloro').selectedStringValue == "CLB
        A_Cloro":
30.         nam1="_CLBA0";
31.         for i in range(CloroNum):
32.             system.tag.addTag(parentPath=path,
33.                 name=municipio+CS+estacion+servicio+"_GR"+grupo+nam1+str(i+1),
34.                 tagType="UDT_INST",
35.                 attributes={"UDTParentType": "Cloro/CLBA"},
36.             )
37. if event.source.parent.getComponent('presion').selected:
38.     if event.source.parent.getComponent('TiposPresion').selectedStringValue == "P
        RBA_Presion":
39.         nam1="_PRBA0";
40.         for i in range(PresionNum):
41.             system.tag.addTag(parentPath=path,
42.                 name=municipio+CS+estacion+servicio+"_GR"+grupo+nam1+str(i+1),
43.                 tagType="UDT_INST",
44.                 attributes={"UDTParentType": "Presion/PRBA"},
45.             )
46.
47. if event.source.parent.getComponent('nivel').selected:
48.     if event.source.parent.getComponent('TiposNivel').selectedStringValue == "NVB
        A_Nivel":
49.         nam1="_NVBA0";
50.         for i in range(NivelNum):
51.             system.tag.addTag(parentPath=path,
52.                 name=municipio+CS+estacion+servicio+"_GR"+grupo+nam1+str(i+1),
```

```
53.         tagType="UDT_INST",  
54.         attributes={"UDTParentType": "Nivel/NVBA"}, )
```

## Anexo B: Código de descripciones de *tags*

```
1. rootFolder = '[default]Municipio/' # The base path of our search
2. collisionPolicy = 'm'
3.
4.
5.
6. def addJSON(tagPath):
7.
8.
9.     splitPath = tagPath.rsplit('/', 1)
10.
11.     #extraer el tooltip
12.     values= str(system.tag.read(tagPath+".Tooltip"))
13.
14.     #a=tagPath.split('/',3)
15.     #pathEstacion='/'.join([a[0],a[1],a[2]])
16.
17.     #extraer el nombre de la estacion
18.     pathEstacion=tagPath.rsplit('/',3)[0]
19.     b=str(system.tag.read(pathEstacion+"/NombreEstacion"))
20.
21.     #extraer el parametro de la instancia
22.     parametrPath=tagPath.rsplit('/',1)[0]+"/Parameters.parametro"
23.     parametrAll=str(system.tag.read(parametrPath))
24.
25.     parametro=parametrAll[1:len(parametrAll)].split(',')[0]
26.     estacion=b[1:len(b)].split(',')[0]
27.     Tooltip=values[1:len(values)].split(',')[0]
28.
29.     description=Tooltip+" "+parametro + " - "+estacion
30.
31.
32.     config = [{'name':splitPath[1], 'Documentation': description}]
33.     system.tag.configure(splitPath[0], config, collisionPolicy)
34.
35.
36.
37. def browseTags(path, filter):
38.
39.
40.     results = system.tag.browse(path, filter)
41.
42.
43.     for result in results.getResults():
44.         if str(result['tagType']) != "Folder" and str(result['tagType']) != "UdtI
nstance" :
45.             addJSON(str(result['fullPath']))
46.
47.
48.         if result['hasChildren'] == True:
49.             browseTags(result['fullPath'], filter)
50.
51.
52. browseTags(rootFolder, {})
```



## Anexo C: Lista de señales del proyecto

Nombre	Ruta	Descripcion
V_CLOR	[default]BRX/01/0001AB/DP01/CLBA01/V_CLOR	Nivel Cloro - La Drova
V_CAUD	[default]BRX/01/0001AB/DP01/CQHL01/V_CAUD	Nivel Caudal - La Drova
V_NIVE	[default]BRX/01/0001AB/DP01/NVBA01/V_NIVE	Nivel Deposito - La Drova
V_PRES	[default]BRX/01/0001AB/DP01/PRBA01/V_PRES	Presión - La Drova
A_CARR	[default]BRX/01/0001AB/GR01/BMMF01/A_CARR	Cuenta Arranques Bomba SAT - La Drova,
A_CHOR	[default]BRX/01/0001AB/GR01/BMMF01/A_CHOR	Cuenta Horas Bomba SAT - La Drova,
E_AVER	[default]BRX/01/0001AB/GR01/BMMF01/E_AVER	Fallo Térmico Bomba SAT - La Drova,
E_FMAR	[default]BRX/01/0001AB/GR01/BMMF01/E_FMAR	Fallo Marcha Bomba SAT - La Drova,
E_FPAR	[default]BRX/01/0001AB/GR01/BMMF01/E_FPAR	Fallo Paro Bomba SAT - La Drova,
E_MARC	[default]BRX/01/0001AB/GR01/BMMF01/E_MARC	Estado marcha SAT - La Drova
T_AUTO	[default]BRX/01/0001AB/GR01/BMMF01/T_AUTO	Automático Bomba SAT - La Drova
T_MANU	[default]BRX/01/0001AB/GR01/BMMF01/T_MANU	Manual Bomba SAT - La Drova
T_MARC	[default]BRX/01/0001AB/GR01/BMMF01/T_MARC	Marcha Bomba SAT - La Drova
T_PARO	[default]BRX/01/0001AB/GR01/BMMF01/T_PARO	Paro Bomba SAT - La Drova
V_INTE	[default]BRX/01/0001AB/GR01/BMMF01/V_INTE	Intensidad Bomba SAT - La Drova
A_CARR	[default]BRX/01/0001AB/GR02/BMMF01/A_CARR	Cuenta Arranques Bomba 1 - La Drova
A_CHOR	[default]BRX/01/0001AB/GR02/BMMF01/A_CHOR	Cuenta Horas Bomba 1 - La Drova
E_AVER	[default]BRX/01/0001AB/GR02/BMMF01/E_AVER	Fallo Térmico Bomba 1 - La Drova,
E_FMAR	[default]BRX/01/0001AB/GR02/BMMF01/E_FMAR	Fallo Marcha Bomba 1 - La Drova
E_FPAR	[default]BRX/01/0001AB/GR02/BMMF01/E_FPAR	Fallo Paro Bomba 1 - La Drova
E_MARC	[default]BRX/01/0001AB/GR02/BMMF01/E_MARC	Estado marcha 1 - La Drova
T_AUTO	[default]BRX/01/0001AB/GR02/BMMF01/T_AUTO	Automático Bomba 1 - La Drova
T_MANU	[default]BRX/01/0001AB/GR02/BMMF01/T_MANU	Manual Bomba 1 - La


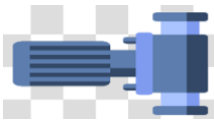

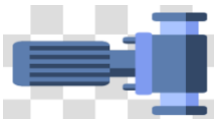

		Drova
T_MARC	[default]BRX/01/0001AB/GR02/BMMF01/T_MARC	Marcha Bomba 1 - La Drova
T_PARO	[default]BRX/01/0001AB/GR02/BMMF01/T_PARO	Paro Bomba 1 - La Drova
V_INTE	[default]BRX/01/0001AB/GR02/BMMF01/V_INTE	Intensidad Bomba 1 - La Drova
A_CARR	[default]BRX/01/0001AB/GR02/BMMF02/A_CARR	Cuenta Arranques Bomba 2 - La Drova
A_CHOR	[default]BRX/01/0001AB/GR02/BMMF02/A_CHOR	Cuenta Horas Bomba 2 - La Drova
E_AVER	[default]BRX/01/0001AB/GR02/BMMF02/E_AVER	Fallo Térmico Bomba 2 - La Drova
E_FMAR	[default]BRX/01/0001AB/GR02/BMMF02/E_FMAR	Fallo Marcha Bomba 2 - La Drova
E_FPAR	[default]BRX/01/0001AB/GR02/BMMF02/E_FPAR	Fallo Paro Bomba 2 - La Drova
E_MARC	[default]BRX/01/0001AB/GR02/BMMF02/E_MARC	Estado marcha 2 - La Drova
T_AUTO	[default]BRX/01/0001AB/GR02/BMMF02/T_AUTO	Automático Bomba 2 - La Drova
T_MANU	[default]BRX/01/0001AB/GR02/BMMF02/T_MANU	Manual Bomba 2 - La Drova
T_MARC	[default]BRX/01/0001AB/GR02/BMMF02/T_MARC	Marcha Bomba 2 - La Drova
T_PARO	[default]BRX/01/0001AB/GR02/BMMF02/T_PARO	Paro Bomba 2 - La Drova
V_INTE	[default]BRX/01/0001AB/GR02/BMMF02/V_INTE	Intensidad Bomba 2 - La Drova
V_PRES	[default]BRX/01/0001AB/GR02/PRBA01/V_PRES	Presión - La Drova
V_PRES	[default]BRX/01/0001AB/GR02/PRBA02/V_PRES	Presión - La Drova
NombreEstacion	[default]BRX/01/0001AB/NombreEstacion	La Drova
V_CLOR	[default]BRX/01/0002AB/CA01/CLBA01/V_CLOR	Nivel Cloro - Monduver
V_CLOR	[default]BRX/01/0002AB/DP01/CLBA01/V_CLOR	Nivel Cloro - Monduver
V_CAUD	[default]BRX/01/0002AB/DP01/CQHL01/V_CAUD	Nivel Caudal - Monduver
V_NIVE	[default]BRX/01/0002AB/DP01/NVBA01/V_NIVE	Nivel Deposito - Monduver
NombreEstacion	[default]BRX/01/0002AB/NombreEstacion	Monduver
A_CARR	[default]BRX/01/0003AB/CU01/BMMF01/A_CARR	Cuenta Arranques Bomba - Pozo Suros
A_CHOR	[default]BRX/01/0003AB/CU01/BMMF01/A_CHOR	Cuenta Horas Bomba - Pozo Suros
E_AVER	[default]BRX/01/0003AB/CU01/BMMF01/E_AVER	Fallo Térmico Bomba - Pozo Suros
E_FMAR	[default]BRX/01/0003AB/CU01/BMMF01/E_FMAR	Fallo Marcha Bomba - Pozo Suros




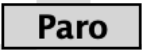




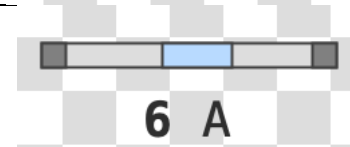









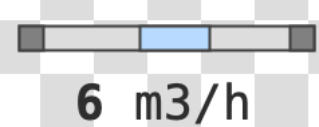
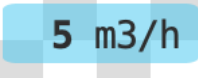






E_FPAR	[default]BRX/01/0003AB/CU01/BMMF01/E_FPAR	Fallo Paro Bomba - Pozo Suros
E_MARC	[default]BRX/01/0003AB/CU01/BMMF01/E_MARC	Estado marcha - Pozo Suros
T_AUTO	[default]BRX/01/0003AB/CU01/BMMF01/T_AUTO	Automático Bomba - Pozo Suros
T_MANU	[default]BRX/01/0003AB/CU01/BMMF01/T_MANU	Manual Bomba - Pozo Suros
T_MARC	[default]BRX/01/0003AB/CU01/BMMF01/T_MARC	Marcha Bomba - Pozo Suros
T_PARO	[default]BRX/01/0003AB/CU01/BMMF01/T_PARO	Paro Bomba - Pozo Suros
V_INTE	[default]BRX/01/0003AB/CU01/BMMF01/V_INTE	Intensidad Bomba - Pozo Suros
V_CAUD	[default]BRX/01/0003AB/CU01/CQHL01/V_CAUD	Nivel Caudal - Pozo Suros
V_NIVE	[default]BRX/01/0003AB/CU01/NVBA01/V_NIVE	Nivel Deposito - Pozo Suros
V_PRES	[default]BRX/01/0003AB/CU01/PRBA01/V_PRES	Presión - Pozo Suros
NombreEstacion	[default]BRX/01/0003AB/NombreEstacion	Pozo Suros
V_CLOR	[default]BRX/01/0004AB/CA01/CLBA01/V_CLOR	Nivel Cloro - Deposito Picayo
V_NIVE	[default]BRX/01/0004AB/DP01/NVBA01/V_NIVE	Nivel Deposito - Deposito Picayo
NombreEstacion	[default]BRX/01/0004AB/NombreEstacion	Depósito Picayo

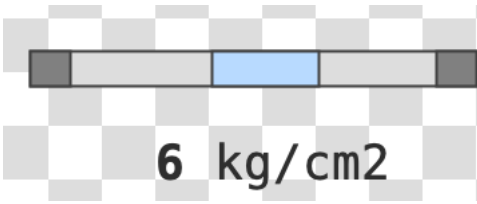
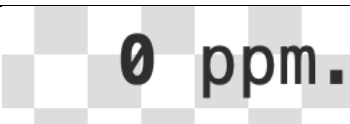

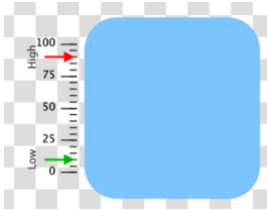






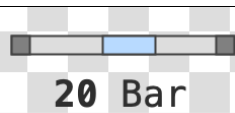




## Anexo D: Lista de *templates* del proyecto

Tipo	Nombre	Template	Descripción
Bomba Básica	Símbolo eléctrico		Es la <i>template</i> que representa la bomba básica en los mapas generales. Dispone de dos estados: Bomba en estado marcha- Bomba en estado paro.
	Símbolo gráfico		Es la <i>template</i> que representa la bomba básica en las ventanas de estaciones. Dispone de dos estados: Bomba en estado marcha- Bomba en estado paro.
Bomba avanzada	Símbolo eléctrico		Es la <i>template</i> que representa la bomba avanzada en los mapas generales. Dispone de tres estados: Bomba en estado marcha- Bomba en estado paro, bomba en fallo.
	Símbolo gráfico		Es la <i>template</i> que representa la bomba avanzada en las ventanas de estaciones. Dispone de dos estados: Bomba en estado marcha- Bomba en estado paro.
	Control Manual      Auto		Controlar estado Manual y

			Automática de la bomba
Control Marcha Paro			Controlar estado de Marcha y Paro de la bomba.
Estado Auto Manual			Indicar el estado de funcionamiento manual automático en una bomba.
Estado Fallo			Indicar el estado de fallos en una bomba.
Estado Marcha Paro			Indicar el estado de funcionamiento Marcha Paro en una bomba.
Alarma Fallo Marcha	 Bomba en Fallo Marcha		Indicar si una bomba tiene un fallo de marcha
Alarma Fallo Paro	 Bomba en Fallo Paro		Indicar si una bomba tiene un fallo de paro
Alarma Fallo Avería	 Bomba en Avería		Indicar si una bomba tiene un fallo de avería
Variable Intensidad 1			Indicar el valor de la intensidad de una bomba.
Variable intensidad 2			Indicar el valor de la intensidad de una bomba.
Variable intensidad 3			Indicar el valor de la intensidad de una bomba.
Variable Numero arranques 1			Indicar el valor de número de arranques de una bomba.
Variable Numero arranques 2			Indicar el valor de número de arranques de una bomba.

	Variable Horas Funcionamiento 1		Indicar el valor de horas de funcionamiento de una bomba.
	Variable Horas Funcionamiento 2		Indicar el valor de horas de funcionamiento de una bomba.
Caudal	Símbolo Eléctrico		Es la <i>template</i> que representa el caudal en los mapas generales.
	Símbolo Grafico		Es la <i>template</i> que representa el caudal en las ventanas de estaciones.
	Variable Valor 1		Indica el valor de la variable caudal.
	Variable Valor 2		Indica el valor de la variable caudal
Cloro	Símbolo		Representa el color en los mapas generales.
	Alarma HH		Indica si la variable cloro presenta un nivel muy alto.
	Alarma H		Indica si la variable cloro presenta un nivel alto.
	Alarma L		Indica si la variable cloro presenta un nivel Bajo.
	Alarma LL		Indica si la variable cloro presenta un nivel muy Bajo
	Variable 1		Indica el valor de la variable cloro.

	Variable 2		Indica el valor de la variable cloro.
	Variable 3		Indica el valor de la variable cloro.
Nivel	Símbolo simple		Representación gráfica de la variable nivel en los mapas generales.
	Símbolo Sparkine		Representa y indica el valor de la variable nivel en las ventanas de estaciones.
	Alarma HH		Indica si la variable nivel presenta un nivel muy alto.
	Alarma H		Indica si la variable nivel presenta un nivel muy alto.
	Alarma L		Indica si la variable nivel presenta un nivel muy alto.
	Alarma LL		Indica si la variable nivel presenta un nivel muy alto.
	Variable 1		Indica el valor de la variable nivel.
Presión	Símbolo		Representación gráfica de la variable presión en los mapas generales y en las ventanas de las estaciones
	Variable 1		Indica el valor de la variable presión.

	Variable 2		Indica el valor de la variable presión.
	Variable 3		Indica el valor de la variable presión.





## Anexo E: Código trazador de variables

```
1. path= system.gui.getParentWindow(event).path
2.
3. #quitar el nombre de la ventana de la ruta
4. path=path[0:13]
5.
6. #prepara las listas
7. tags=[]
8. name=[]
9. fullPath=[]
10. value=[]
11. vsource=[]
12. histo=[]
13. alarm=[]
14. Description=[]
15.
16. #funcion para examinar los tags en la ruta
17. def browseTags(path):
18.
19.     # Call the browse function
20.     results = system.tag.browse(path)
21.
22.     # Loop through every item in the results and print it
23.     for result in results.getResults():
24.
25.         tags.append(result)
26.         # If the item has children, call the function to repeat the process but s
27.         tarting at the child.
28.         if result['hasChildren'] == True:
29.             browseTags(result['fullPath'])
30.
31. #llamar a la funcion
32. browseTags(path)
33.
34. #de los tags encontrados, ignoramos Datatypes y carpets. y extraeremos informacion
35. necesaria
36. for i in range(0,len(tags)):
37.     if str(tags[i]["tagType"]) == 'AtomicTag':
38.         fullPath.append(tags[i]["fullPath"])
39.         Description.append(system.tag.read(str(tags[i]["fullPath"])+".Documentati
40. on"))
41.         name.append(tags[i]["name"])
42.         value.append(tags[i]["value"])
43.         vsource.append(tags[i]["valueSource"])
44.         attributes=list(tags[i].get("attributes"))
45.         #for j in range(0,len(attributes)):
46.         if 'NodeAttribute[history]' in str(attributes):
47.             histo.append("Si")
48.         else:
49.             histo.append("No")
50.
51.         if 'NodeAttribute[alarm]' in str(attributes):
52.             alarm.append("Si")
53.         else:
54.             alarm.append("No")
55.
56. #Guardar los datos en una tabla
57. hdr = ['Nombre', 'Ruta', 'Descripcion', 'Valor', 'Tipo', 'Historicos?', 'Alarmas?']
58. fullIDS = zip(name,fullPath,Description,value,vsource,histo,alarm)
59. data=system.dataset.toDataSet(hdr,fullIDS)
```

```
59. #Abrir la ventana de trazador de variables, y pasarle el la tabla como parametro
60. window = system.nav.openWindowInstance('Popup/Trazador Variables', {'Data' : data
    })
61. system.nav.centerWindow(window)
```

## Anexo F: Código Integración de *dataloggers*

```
1. import csv
2. import itertools
3.
4. def readFile(filename):
5.     with open(filename, "rb") as csvfile:
6.         datareader = csv.reader(csvfile)
7.         next(datareader) # ignorar el encabezado
8.         for row in datareader:
9.             yield row
10.
11. def chunks(iterable, size=1000):
12.     iterator = iter(iterable)
13.     for first in iterator:
14.         yield itertools.chain([first], itertools.islice(iterator, size - 1))
15.
16. prov = 'EntHistory'
17. path = system.file.openFile("csv")
18.
19.
20. fullDS = []
21. for chunk in chunks(readFile(path)):
22.     vals = []
23.     paths = []
24.     quals = []
25.     ts = []
26.
27.     for row in chunk:
28.         filas=row
29.
30.         parte1=filas[0]
31.         filasplit=parte1.split(';')
32.         nombre=filasplit[0]
33.         FechaHora=filasplit[1]
34.         ValorEntero=filasplit[2]
35.
36.         #extrear fecha
37.         fecha=filasplit[1][0:10]
38.         mes=int(fecha.split('/')[0])-1
39.         dia=int(fecha.split('/')[1]) # 0 es enero
40.         anio=int(fecha.split('/')[2])
41.
42.         #extrear hora
43.         if len(filasplit[1])==14:
44.             hora=filasplit[1][11:15]
45.             h=int(hora.split(':')[0])
46.             minu=int(hora.split(':')[1])
47.         else:
48.             hora=filasplit[1][11:16]
49.             h=int(hora.split(':')[0])
50.             minu=int(hora.split(':')[1])
51.
52.         #extraer valor
53.
54.         valor=float(filasplit[2]+'.'+filas[1].split(';')[0])
55.
56.
57.         #extraer ruta
58.         name1=nombre.split('.')
59.         name2=name1[0].split('_')
60.         name3=name2[0][0:3]
61.
```

```

62.         ruta=name2[0][0:3]+'/' +name2[0][3:5]+'/' +name2[0][5:11]+'/' +name2[1]+'/' +
        name2[2]+'/' +name1[1]
63.         date = system.date.getDate(anio, mes, dia)
64.         gd = system.date.setTime(date, h, minu, 00) #segundos?
65.
66.
67.
68.         paths.append(ruta)
69.         vals.append(valor)
70.         quals.append(192)
71.         ts.append(gd)
72.         prov="BaseDeDatos"
73.         TagProvider="default"
74.         system.tag.storeTagHistory(prov,TagProvider,paths,vals,quals,ts)
75.     hdr = ['tag', 'val', 'qual', 'date']
76.     fullDS = zip(paths,vals,quals,ts)
77.     event.source.parent.getComponent('Power Table 3').data=system.dataset.toDataSet(h
        dr,fullDS)

```